Scalable User Interfaces for Collaborative Extended Reality Environments

Vom Fachbereich Informatik der
Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau
zur Verleihung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation
von

Vera Marie Memmesheimer

Datum der wissenschaftlichen Aussprache
21. November 2025
Dekan
Prof. Dr. Christoph Garth
Vorsitz der Promotionskommission
Prof. Dr.-Ing. Jörg Dörr
Berichterstatter
Prof. Dr. Achim Ebert (RPTU)
Prof. Dr.-Ing. Jan C. Aurich (RPTU)
Prof. Dr. Bahram Ravani (University of California Davis, USA)

DE-386

Für O.R., O.M. und O.L.

Abstract

Extended Reality (XR) unlocks new opportunities for interacting with spatial content, supporting both individual users and diverse collaborative settings. Different degrees of virtuality like Augmented/Mixed Reality (AR/MR) and Virtual Reality (VR) as well as different devices like head-mounted displays (HMDs) and handheld displays (HHDs) offer distinct benefits for different use cases. Despite the immense interest XR has sparked in numerous domains, it is rarely used in practice where scalability limitations outweigh XR's intrinsic potential. Since users are typically involved in multiple use cases, leveraging technology-specific benefits requires switching between use cases and XR technologies. However, existing user interfaces (UIs) impede these transitions because they are tailored to specific use cases. Thus, this dissertation is concerned with the development of highly scalable UIs which facilitate switching between XR applications that differ in technology and number of users.

First, Scalable Extended Reality (XR^S) is introduced as a novel concept for XR spaces which scale across different devices, degrees of virtuality, and varying numbers of potentially distributed users. A research agenda addressing the barriers to the realization of XR^S is established based on an extensive compilation of related research. As an initial step towards XR^S and as the basis for this dissertation the XR^S framework is developed.

By sharing spatial content among collaborators, XR could overcome a key limitation of conventional videoconferencing tools. However, collaboration support features face scalability issues when different XR technologies are used in large groups. A particular challenge concerns the accurate representation of HHD users. In response, the dissertation presents new insights from a detailed study on how humans interact with HHDs across different display sizes, display orientations, and body poses. Extending these results, mechanisms for individually activating the visibility of awareness cues are designed to reduce visual overload in large groups.

Next, interaction techniques scaling with devices and degrees of virtuality are presented. Starting with MR-HHDs, a unified paradigm for object translation and rotation is developed and evaluated. By combining tablet movement and peripheral touch input its minimalist design overcomes key issues of prior methods. Extending HMDs with a tablet controller using this paradigm yields consistent interaction with MR-HHDs, MR-HMDs, and VR-HMDs. In a comparative study, this solution outperformed state-of-the-art methods and revealed high scalability. Eventually, the novel UIs are implemented and evaluated for robot control and factory layout planning, showcasing their practical applicability.

Zusammenfassung

Extended Reality (XR) eröffnet neue Möglichkeiten, mit räumlichen Inhalten zu interagieren und unterstützt sowohl einzelne Nutzerinnen und Nutzer als auch verschiedenste kollaborative Umgebungen. Dabei bieten unterschiedliche Virtualitätsgrade wie Augmented/Mixed Reality (AR/MR) und Virtual Reality (VR) sowie Geräte wie Head-mounted Displays (HMDs) und Handheld Displays (HHDs) spezifische Vorteile für verschiedene Anwendungsfälle. Obwohl XR Technologien in zahlreichen Bereichen enormes Interesse geweckt haben, finden sie bisher nur selten praktische Anwendung, weil die begrenzte Skalierbarkeit das intrinsische Potenzial von XR relativiert. Da Nutzerinnen und Nutzer in der Regel in mehrere Anwendungsfälle involviert sind, erfordert die Ausschöpfung der technologiespezifischen Stärken nicht nur den Wechsel zwischen Anwendungsfällen sondern auch zwischen XR Technologien. Existierende User Interfaces (UIs) sind jedoch meist auf spezielle Anwendungsfälle zugeschnitten und erschweren diese Übergänge. Die vorliegende Dissertation befasst sich daher mit der Entwicklung hoch skalierbarer UIs, die einen nahtlosen Wechsel zwischen XR Anwendungen ermöglichen – insbesondere dann, wenn diese verschiedene Technologien einsetzen und von einer unterschiedlichen Anzahl von Personen genutzt werden.

Zunächst führt die Dissertation Scalable Extended Reality (XR^S) als neues Konzept für XR Umgebungen ein, die hinsichtlich variierender Geräte, Virtualitätsgrade sowie der Anzahl potenziell räumlich verteilter Nutzerinnen und Nutzer skalieren. Ausgehend von einer umfassenden Analyse des bisherigen Forschungsstandes wird anschließend eine Agenda zur Überwindung der Realisierungsbarrieren von XR^S erstellt. Als erster Schritt in Richtung XR^S und als Grundlage dieser Dissertation wird das XR^S Framework entwickelt.

Dank der Möglichkeit, kollaborativ mit räumlichen Inhalten zu interagieren, könnte XR eine zentrale Schwäche herkömmlicher Videokonferenz-Tools schließen. Existierende Lösungen zur Unterstützung von Kollaboration in XR unterliegen in großen Gruppen und unter dem Einsatz verschiedener XR Technologien jedoch mangelnder Skalierbarkeit. Eine besondere Herausforderung besteht hier in der akkuraten Repräsentation von HHD Nutzerinnen und Nutzern. Die Dissertation befasst sich mit dieser Thematik im Rahmen einer detaillierten Studie, welche neue Erkenntnisse über die Interaktion mit HHDs unterschiedlicher Bildschirmgrößen, Bildschirmorientierungen und Körperhaltungen liefert. Um visuelle Überlastung in großen Gruppen zu vermeiden, werden ergänzend Mechanismen zur individuellen Aktivierung von sogenannten Awareness Cues entworfen.

Anschließend werden Interaktionstechniken präsentiert, die mit verschiedenen Geräten und Virtualitätsgraden skalieren. Beginnend mit MR-HHDs wird ein neues Interaktions-

paradigma zur Objektverschiebung und -rotation entwickelt und evaluiert. Durch sein minimalistisches Design, das Tabletbewegung und periphere Toucheingabe kombiniert, werden zentrale Einschränkungen bisheriger Methoden überwunden. Die Ergänzung von HMDs durch eine Tabletsteuerung, die das gleiche Paradigma einsetzt, ermöglicht eine konsistente Interaktion über MR-HHDs, MR-HMDs und VR-HMDs hinweg. In einer vergleichenden Studie übertraf diese Lösung State-of-the-Art-Methoden und zeigte eine hohe Skalierbarkeit. Schließlich werden die neuen UIs für Robotersteuerung und Fabriklayoutplanung implementiert und evaluiert, um ihre praktische Anwendbarkeit zu demonstrieren.

Acknowledgments

Over the course of my doctoral research, I was fortunate to receive support from many people and would like to thank everyone who contributed, directly or indirectly, to the completion of this dissertation.

First and foremost, I am deeply grateful to my supervisor, Prof. Dr. Achim Ebert, for giving me the opportunity to pursue a doctoral degree in the HCI lab and for providing invaluable guidance throughout my academic journey.

In addition, my sincere gratitude goes to Prof. Dr.-Ing. Jan C. Aurich for providing valuable feedback on my research in the context of IRTG 2057 and beyond.

I would also like to express my deep appreciation to Prof. Dr. Bahram Ravani whose continuous support and broad expertise have shaped my work. Thanks to his invitation, my research stays in the Department of Mechanical and Aerospace Engineering at UC Davis were made possible – experiences that I reflect on with immense gratitude.

I am grateful to the chair of my PhD committee, Prof. Dr.-Ing. Jörg Dörr, for investing time and effort in the examination process as well as to Prof. Dr. Christoph Garth, Prof. Dr. Hans Hagen, Prof. Dr. Heike Leitte, and Prof. Dr. Katharina A. Zweig for generously sharing their knowledge and advice throughout the past years.

Many thanks go to our fantastic team assistant and my office neighbor, Janine Mertel, for her consistent availability and support in administrative matters. I would also like to thank Roger Daneker and Mady Gruys for their help and kind words.

I wish to thank all my former and current colleagues in the HCI and VIS teams for creating such a supportive and enjoyable working environment. Looking back, I am left with so many unforgettable memories – merci! Beyond HCI and VIS, I had the pleasure of meeting researchers from other fields including Mechanical Engineering, Robotics, as well as Psychology and deeply value the interdisciplinary discussions that fostered instrumental collaborations. Across all these groups, I am grateful for the personal bonds that have formed beyond academia and the people who have become treasured parts of my life.

Throughout my time in the HCI lab, it was my pleasure to supervise about 30 theses / projects and discuss research topics with many students – some of whom later became co-authors and colleagues. Special thanks to Kai J. Klingshirn, Sofie M. Schwenkreis, Cindy Herold, and Hansika Subbaraj who worked with me over a long period of time.

My profound thanks go to all my co-authors, whose diverse skills and perspectives have

greatly enriched my research.

For me, one of the most rewarding aspects of HCI research has always been to observe how humans interact with novel user interfaces. Therefore, I would like to extend my sincere thanks to everyone who participated in my experiments.

I am thankful for the resources and opportunities provided by the Bund-Länder-Initiative Innovative Hochschule – BMFTR / GWK as well as by the German Research Foundation (DFG). It has been a privilege to be a part of the Offene Digitalisierungsallianz Pfalz (ODPfalz-II) and the International Research Training Group (IRTG 2057) which allowed me to meet and collaborate with researchers from different fields and countries.

Apart from all of this, I am incredibly grateful to my family and the friends who crossed my path, somewhere between kindergarten and today. Thank you for accompanying me on this journey and enriching my life in so many ways.

My deepest and most special thanks go to my parents and my brother whose support has been endless and unconditional. It is a priceless privilege to know that I can always count on you.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	two-dimensional
3D	three-dimensional
App	Application
AR	Augmented Reality
AV	Augmented Virtuality
CAD	Computer-Aided Design
DOF	Degree of Freedom
FLP	Factory Layout Planning
GUI	Graphical User Interface
HCI	Human Computer Interaction
HHD	Handheld Display
HMD	Head-mounted Display
MR	Mixed Reality
REQ	Requirement
TAM	Technology Acceptance Model
TCT	Task Completion Time
TTF	Task Technology Fit
UA	User Acceptance
UI	User Interface
VR	Virtual Reality
WIM	Worlds In Miniature
WIMP	Windows Icons Menus Pointers
XR	Extended Reality
XR^{S}	Scalable Extended Reality

Chapter 1

Introduction

1.1 Motivation

Extended Reality (XR) promises to transform the way humans interact with digital content and has sparked immense interest across diverse domains. Its technological variety encompasses both Virtual Reality (VR) which allows immersing users in entirely computer-generated three-dimensional (3D) environments as well as Augmented/Mixed Reality (AR/MR) which blends reality with virtual elements and thus allows simultaneous interaction with physical and virtual elements. Users can access these environments with different hardware including ubiquitous devices like handheld displays (HHDs) as well as head-mounted displays (HMDs) which are specifically designed for XR applications. The individual advantages offered by this variety of technologies open up a vast field of potential XR applications.

For example, XR can support product design (e.g., in the automotive or aerospace industry) as it allows stakeholders to interact with real-size 3D product models. Thereby, VR environments can allow reviewing entirely virtual product models in early development stages whereas in AR/MR physically existing parts of a product can be augmented with virtual elements. Similarly, XR can deliver support in different planning and building stages of private, public, and industrial properties.

By immersing users in virtual replicas of physically existing (or non-existing) environments, XR can enhance the availability of training conditions. Applications which can benefit from minimized safety issues and supervision resources in such virtual environments include training maintenance, repair, or assembly tasks, as well as emergency pro-

tocols, or surgical procedures.

Furthermore, remote assistance tasks have been identified to particularly benefit from XR. Here, a local worker can receive assistance from a remote expert who is immersed in a virtual replica of the local environment. The remote expert can then add visual augmentations to the shared space to guide the local worker in completing a task. With this use case, XR appears to be able to close a meaningful gap in existing collaboration-support tools: The Covid-19 pandemic has increased the evolution and adoption of videoconferencing tools which have replaced many – yet not all – face-to-face meetings. While common platforms enable the exchange of two-dimensional (2D) content via screen sharing, they are not applicable to collaborative tasks which require sharing spatial data. Here, the application of XR appears particularly promising.

The immense potential assigned to XR has fostered the development of novel hardware as well as research in different areas such as enhanced interaction, scene generation, and avatar representations. However, XR's actual implementation in practical settings remains limited. While the variety of XR technologies offers individual strengths from which users can benefit in different use cases, this technological variety also poses challenges to the development of appropriate user interfaces (UIs). To leverage technology-specific benefits and allow users to choose XR technologies solely based on their preferences and the specific use case, scalable UIs are needed which facilitate switching between VR and MR environments as well as between HMDs and HHDs.

The lack of such scalable UIs for XR technologies potentially impedes their application in the real world and motivates the research presented in this dissertation.

1.2 Contributions

The primary insight driving this dissertation is that different XR technologies offer individual strengths from which users can benefit in different use cases, making seamless switching between the technologies a key requirement. The current use case driven development of XR applications results in UIs for specific hardware, tasks, and numbers of users. As such, using XR technologies for multiple tasks requires temporal and cognitive efforts to adapt to new systems. To leverage each technology's potential across different use cases while maintaining usability, this dissertation follows a comprehensive approach to design UIs that stay consistent when switching between XR technologies

independently of the number of potentially distributed users. The main contributions can be summarized as follows:

Scalable Extended Reality. First, XR^S is introduced as a concept for XR spaces that provide scalability with respect to the degree of virtuality (MR/VR), devices (HHDs/HMDs), and the number of users (single users, co-located and distributed collaborators). Results from the identified related research fields are compiled and current barriers to the realization of XR^S are summarized in a research agenda. As an initial step towards XR^S, a corresponding framework is developed. To this end, abstract use cases leveraging XR's key benefits are derived from a review of potential XR applications. Based on these abstract use cases, functional and non-functional requirements are formulated and translated into the framework which provides several scalability enhancements: Multiple on-site and off-site users can access a joint XR^S space through customized MR-UIs or VR-UIs and then reference or manipulate real and virtual scene components. Thereby, the integration of a robotic system is proposed to enable manipulation of distant, large, heavy, or hazardous objects. The designed framework can be adapted to any specific application which can be described by (combinations of) the abstract use cases and builds the basis for the scalable UIs developed in this dissertation.

Scalable Collaboration Support Features. In collaborative XR settings, each user should be provided with appropriate representations of the other collaborators and their activities. However, scalability issues such as incorrect user representations and visual overload can occur when existing collaboration support features are applied across different XR technologies and varying group sizes. In contrast to HMDs, it remains unclear how an HHD user's viewing direction can be determined precisely. Addressing this question, this dissertation presents new insights on how users interact with HHDs which differ in the size and orientation of the display and are used in different body poses. Investigations regarding the front camera's accessibility during interaction delivered positive results, indicating the potential to integrate face tracking to enhance MR-HHD user representations in collaborative settings. On top of that, it is shown that a ray originating from the device center is, in general, an appropriate indicator for the user's viewing direction, but its accuracy is affected by device orientation, viewing direction, and distance. Based on these findings, recommendations for HHD user representations are derived.

Moreover, this dissertation presents the design of mechanisms for enabling and disabling visual cues to prevent visual overload in large groups. To this end, diverse co-located and distributed collaboration styles involving HMDs and HHDs were taken into consideration. Based on natural cooperation paradigms, mechanisms for automatically adapting the visibility of cues are designed. Thus, users can individually enable and disable cues that display the other collaborators' activities. At the same time, they can draw attention to themselves and take influence on their collaborators' fields of view.

Scalable Interaction Techniques. A set of novel object manipulation techniques for MR-HHDs, MR-HMDs, and VR-HMDs is presented. A central contribution here is the design and development of *Move'n'Hold* an interaction technique for MR-HHDs that applies the same paradigm for translating and rotating virtual objects and addresses usability issues present in existing methods. It provides simplicity through a manageable set of input modalities (i.e., device movement and peripheral touch) and allows always holding the device with both hands to reduce fatigue and occlusion issues. *Move'n'Hold* enables direct manipulation by mapping device movement to virtual objects while touch is applied on the left display side. At the same time, automated repetitions of these direct manipulations can be started or stopped when touch is added or released on the right display side. In this way, *Move'n'Hold* provides individual combinations of natural manipulation for small, precise movements and continuous manipulation for large, coarse movements. The conducted user studies revealed *Move'n'Hold* as an intuitive and easy-to-learn spatial interaction technique that provides scalability in terms of the distance, direction, complexity, and speed of manipulation. Furthermore, it supports different user preferences and interaction styles. To maintain consistency when users have to switch between XR technologies, the design of *Move'n'Hold* was extended for MR-HMDs and VR-HMDs. Thereby, virtual objects seen through the HMD can be manipulated with a tablet controller that implements the same interaction paradigm as *Move'n'Hold* for MR-HHDs. The participants of a study who used *Move'n'Hold* and state-of-the-art methods for object manipulation with MR-HHDs, MR-HMDs, and VR-HMDs preferred *Move'n'Hold* and rated it to be easier to relearn. The results further show that it reduced workload, improved usability, and provided more cross-device benefits, facilitating switching between devices and degrees of virtuality.

Practical Applications. Through two practical examples, the dissertation demonstrates the applicability of the developed UIs. Robot control is enabled through a MR-HHD-UI that allows manipulating virtual replicas of physical objects and commanding a robot to perform the same operation with the physical object. In a user study, the proposed UI turned out to be a powerful tool that successfully combines the capabilities of humans and robots. In the context of the XR^S framework, this addresses the robotic system for manipulating distant, large, heavy, or hazardous objects. Furthermore, XR^S is applied across different stages of a factory layout planning process and evaluated in a pilot study. To this end, object manipulation based on *Move'n'Hold* is implemented and evaluated for a multi-user MR-HHD application where the arrangement of factory units can be optimized regarding criteria like transportation intensity in the conceptual planning stage and a VR-HMD application which allows optimizing single factory units for example regarding walking distances and ergonomics in the detailed planning stage.

1.3 Structure

This dissertation is structured into seven chapters. Following the introduction, Chapter 2 introduces the relevant terminology and background information for this dissertation. Thus, fundamentals on XR technologies, collaborative XR settings, and interaction in XR are provided. Furthermore, it summarizes learnings drawn from research on user acceptance models which influenced the research focus of this dissertation along with fundamentals on cognitive load and usability.

Chapter 3 introduces the concept of XR^S along with an overview of XR's potential fields of application and a summary of research relevant to the realization of XR^S (i.e., collaboration support features, scene generation, and interaction techniques). Based on the literature summarized, barriers to the realization of XR^S are identified and translated into a research agenda. Setting the stage for XR^S, the second part of the chapter presents a framework for XR^S which serves as the basis for this dissertation.

Together, Chapter 2 and 3 provide information which is relevant across multiple of the following chapters. Related research and aspects which are relevant to a specific chapter are summarized in separate sections of Chapter 4, 5, and 6 respectively.

Chapter 4 is focused on collaboration support features scaling with different degrees of

virtuality, devices, and group sizes. To address identified scalability issues, the behavior of MR-HHD users is investigated as a foundation for enhanced user representations and mechanisms are designed to individually adapt the visibility of awareness cues.

Chapter 5 presents scalable interaction techniques which stay consistent as users switch between different degrees of virtuality and devices. To this end a novel object manipulation paradigm for MR-HHDs is presented and evaluated in a user study. Based on the insights gained, the paradigm is then adapted and extended for MR-HMDs and VR-HMDs. In a second user study, the novel set of consistent interaction techniques is compared to a set of state-of-the-art techniques.

Chapter 6 presents two practical applications of the developed UIs. The first part of the chapter shows how a robotic system can be controlled through a MR-HHD-UI in a pick and place task. In the second part of the chapter, the developed scalable UIs are applied across different stages of a factory layout planning process. To this end, a multi-user MR-HHD application for the conceptual planning phase and a VR-HMD application for the detailed planning phase are developed and evaluated in a pilot study.

Eventually, Chapter 7 provides a brief discussion of the results along with an outlook on further applications and future research opportunities.

Parts of this dissertation present work which has been previously published and is based on collaborations with other researchers and students who have either completed a thesis/project or worked in our lab under my supervision. At the end of each chapter, the respective contributors are named as co-authors in the list of publications containing presented material. In appreciation of my collaborators, I will use the academic *we* for describing the research work presented in this dissertation.

Chapter 2

Background

In this chapter we introduce background information which is relevant to multiple of the following chapters along with the terminology used in this dissertation. The chapter is structured into four parts. First, an overview of Extended Reality (XR) technologies including different degrees of virtuality and devices is given. Next, it is outlined how different collaboration styles can benefit from XR and how information on scene understanding and user behavior can be shared among collaborators. This is followed by a description on how users of XR technologies can interact with virtual elements. Thereby, essential differences between non-spatial and spatial interaction are explained along with examples of common interaction approaches. Finally, learnings from user acceptance models which influenced the research focus of this dissertation are summarized together with fundamentals on cognitive load and usability.

2.1 Extended Reality Technologies

While concepts for virtually augmented and purely virtual environments date back to the last century [45], a uniform terminology for such environments is still missing. The terminology used in this dissertation is based on the so-called *Reality-Virtuality-Continuum* which was introduced in the 1990s by Milgram et al. [115, 116]. The continuum ranges from real physical environments to entirely computer-generated virtual environments, which are today often referred to as *Virtual Reality (VR)*. The continuum uses the term *Mixed Reality (MR)* to describe environments which contain both real and virtual elements. As such, the term MR was originally introduced as an umbrella term for *Augmented Reality (AR)* and *Augmented Virtuality (AV)*. Thereby, AR refers to real environ-

ments which are augmented with virtual elements and AV refers to virtual environments that are augmented with real elements.

Since then, the environments encompassed by the Reality-Virtuality-Continuum have been the subject of an increasing number of research projects. Compared to AR, considerably less research has been conducted on AV such that the terms AR and MR are today often used synonymously for virtually augmented real environments. When distinguishing between the two, AR frequently refers to simple virtual overlays in physical environments, while MR rather refers to virtual objects that are spatially integrated into the physical environment. An umbrella term which has emerged and is now frequently used to refer to the environments along the Reality-Virtuality Continuum is *Extended Reality (XR)* (sometimes also *Cross Reality*).

Apart from this, the term *Mediated Reality* [106] was introduced to refer to the modification of our visual perception of reality, for example by augmenting, diminishing, or altering it in other ways. In this context, term *Diminished Reality* has also emerged, referring to environments in which physically existing components are removed from our perception of reality [45].

This dissertation is focused on MR, VR, and XR as defined below:

Mixed Reality (MR) refers to environments which blend real and virtual content. Thus, the user can see virtual elements which are seamlessly integrated into a physical environment.

Virtual Reality (VR) refers to entirely computer-generated environments. Thus, the user is immersed in a virtual scene in which the physical world is no longer visible.

Extended Reality (XR) serves as an umbrella term for MR and VR. Thus, it covers spaces which range from physical environments that integrate different amounts of virtual elements to entirely virtual environments.

In the context of MR and VR, we also refer to *different degrees of virtuality*, i.e., MR offers lower degrees of virtuality than VR.

MR and VR scenes can be accessed with *head-mounted displays (HMDs)* which project stereoscopic images in front of the user's eyes and update them according to the user's head movements. As such, today's HMDs build up on pioneering work of Sutherland [163].

Regarding HMDs for MR, it can be distinguished between optical see-through devices and video see-through devices [45]. Optical see-through devices like the first and second generation of Microsoft HoloLens have a transparent display such that the user can see the physical world through the display and virtual augmentations are directly projected in front of the user's eyes. On the contrary, video see-through devices such as the Apple Vision Pro are fully closed and provide the user with a live video stream of the physical surroundings which is augmented with virtual elements. HMDs for VR such as the HTC VIVE Pro are fully closed and designed for immersing the user in an entirely computer-generated scene.

An alternative access to MR is offered by *handheld displays (HHDs)* such as smartphones or tablets [45]. Such MR-HHDs work in a similar way as video see-through MR-HMDs. Thereby, the MR-HHD displays the real environment as captured through the device's camera and augments it with virtual elements.

Both HMDs and HHDs offer different advantages and disadvantages. Compared to HMDs, HHDs provide less immersion and screen space. On the contrary, HHDs provide greater availability as they are cheaper and already widely used in both private and professional contexts.

Beyond HHDs and HMDs, XR can also be realized through stationary projection-based systems [45]. Powerwalls offer large displays onto which stereoscopic images are projected to display 3D images or videos in stationary VR settings. CAVE systems [32] are based on the same concept but combine multiple of these walls to create a more immersive setup. In the context of MR settings, the term Spatial Augmented Reality (SAR) [45, 13] refers to physical settings which are superimposed with virtual augmentations through projectors. However, such stationary setups can be more expensive and are thus rather considered for specialized than mainstream use.

This dissertation is focused on three types of XR technologies (i.e, MR-HHDs, MR-HMDs, and VR-HMDs) as defined below.

Mixed Reality Handheld Displays (MR-HHDs) are smartphones or tablets displaying the real environment as captured through the device camera together with virtual augmentations.

Mixed Reality Head-mounted Displays (MR-HMDs) are head-worn devices which allow users to see virtual objects that augment the real environment,

either through a transparent display (optical see-through) or through a live video stream (video see-through).

Virtual Reality Head-mounted Displays (VR-HMDs) are head-worn devices which immerse the user in a completely computer-generated scene.

When describing XR applications it is important to distinguish between the physical existence of an object and its visual appearance. Regarding the physical existence of an object, we use the terms *real components* (to refer to a physically existing object) and *virtual components* (to refer to purely virtual objects). For example, an MR application for reviewing a car design can extend the physically existing parts (real components) of the car with virtual augmentations displaying the missing parts (virtual components). If such a MR user is joined by a remote collaborator, this collaborator can be provided with a corresponding VR application. The VR application then displays virtual replicas of the real components and integrates them with the virtual components. In such distributed settings, a real component has a different visual appearance for the MR user (physical appearance) and the VR user (virtual appearance). However, when interacting with this component both the MR user and the VR user are referring to the identical scene component.

This distinction is crucial when referring to specific scene components in such collaborative settings. In many other contexts, however, the physical existence of an object is not relevant and we just want to express that the visual appearance of a component is virtual. In these cases, we will use the term *virtual element* to refer to both purely virtual components as well as to virtual replicas of real components.

Furthermore, we use the terms *static* and *dynamic* to indicate if scene components are meant to change their position or orientation during runtime.

The corresponding terms as used in this dissertation are defined as follows.

Real components are physically existing objects.

Virtual components are exclusively virtual objects.

Virtual elements refer to the virtual appearance of these components; they encompass virtual components and virtual replicas of real components.

Static components are not meant to change their position or orientation while using the XR application.

Dynamic components can change their position or orientation while using the XR application.

Furthermore, we distinguish between on-site and off-site users. We use the term *on-site users* for those users who access the XR application from the actual working environment. For example, the location of physically existing parts of a virtually augmented prototype. On the contrary, we refer to users as *off-site users* if they are located at a site where no real components exist. For example, they can access an XR application which includes virtual replicas of items that physically exist elsewhere.

On-site users are located at the same site as real components which are part of the XR application.

Off-site users are located at a site where no real components exist.

2.2 Collaboration in Extended Reality

Digital tools for supporting collaboration have been developed for a long time and are being used in many workplaces. The Covid-19 pandemic has further boosted the consumer interest [172] in these tools, their usage [38] as well as research in this area [36]. While tools like Zoom and Microsoft Teams are well-suited for collaborative use cases in which 2D content is exchanged via screen sharing, options for sharing spatial content is limited.

In this context, applying XR seems predestined as it does not only allow displaying but also sharing and thus collaboratively interacting with 3D content. In fact, it allows sharing both 3D content which is available only in digital form as well as virtual 3D replicas of content which is physically present elsewhere. Thus, XR can be of benefit for both physically distributed and co-located collaborators.

Forms of collaboration can be categorized with respect to time and place [85]. The collaborators can be at the same place (*co-located*) or at different places (*distributed*). In each case, collaboration can either take place at the same time (*synchronously*) or at different times (*asynchronously*). This dissertation is focused on synchronous settings of XR-supported co-located or distributed collaboration. Thereby, different XR technologies may be chosen depending on the collaborators' locations and the specific use case.

Previous research has outlined that XR technologies can support diverse collaborative

settings [134]. For example, MR-HHDs [184, 68] and MR-HMDs [193] have been employed in co-located collaborative settings. Such MR settings allow collaboratively reviewing spatial data while also seeing each other. In distributed settings, VR-HMDs are often employed to immerse a remote collaborator in a virtual reconstruction of the physical scene of a local user wearing an MR-HMD [5, 91, 92, 99, 138, 140, 167, 23, 136, 193]. On top of that, collaborative XR systems have been proposed which involve not only different degrees of virtuality (i.e., VR and MR) but also different devices (i.e., HHDs and HMDs) [133, 57, 96].

This dissertation considers nine different synchronous collaboration styles (see Fig. 2.1). If two collaborators use different devices, two collaboration styles need to be considered (e.g., *MR-HHD user sees VR-HMD user* and *VR-HMD user sees MR-HHD user*) which pose different requirements to tracking and visualizing user behavior.

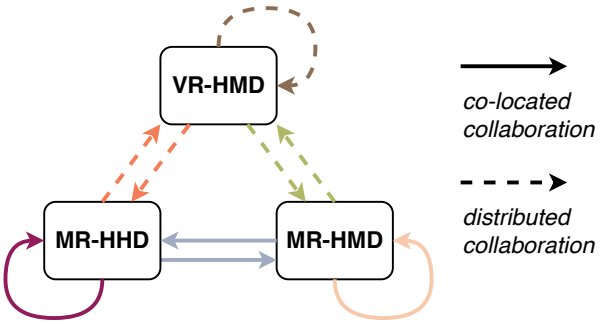


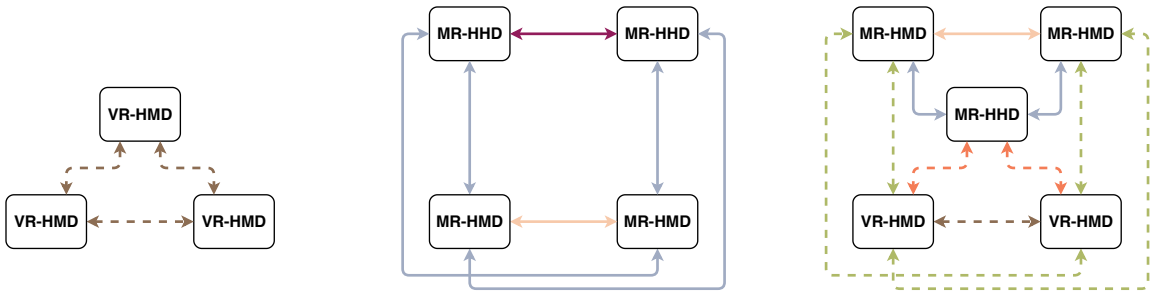
Figure 2.1: The nine arrows represent the nine collaboration styles which are considered in this dissertation. For example, the arrow pointing from MR-HHD to VR-HMD serves as a reference for describing how a MR-HHD user sees a VR-HMD user, the opposite arrow refers to the way a VR-HMD user sees a MR-HHD user. Self-referencing arrows describe how a collaborator sees another collaborator using the same XR technology.

Thereby, the term *distributed collaboration* is used for all kinds of interaction which take place among a VR-HMD user and any other collaborator. This also includes collaboration between two VR-HMD users. On the contrary, collaboration among MR technologies is referred to as *co-located collaboration*.

Co-located collaboration takes place between collaborators who are physically located at the same site and use MR technologies.

Distributed collaboration takes place when collaborators are physically distributed or when at least one collaborator uses a VR-HMD.

The terms co-located and distributed collaboration are used to refer to the collaboration style of one pair of collaborators. That is, a collaborative setting is described by a set



(a) Three distributed collaborators using VR-HMDs.

(b) Four co-located collaborators using MR-HHDs and MR-HMDs.

(c) Three co-located collaborators using MR-HMDs and a MR-HHD are joined by two remote collaborators using VR-HMDs.

Figure 2.2: Examples of collaborative settings involving different XR technologies.

of collaboration styles and the involved XR technologies. Three examples for such a collaborative setting are illustrated in Fig. 2.2.

XR-supported collaboration tools consist of multiple components. Apart from suitable networking solutions which synchronize data among the collaborators and handle access control to scene components, a joint scene understanding in distributed settings requires appropriate representations of the on-site environment. In this context, virtual replicas of the on-site environment were generated with 360-degree [167, 193, 140, 99] or RGB-D cameras [5, 103]. To enable effective collaboration, spatial information describing not only the task domain but also user behavior must be conveyed among the collaborators in an appropriate manner. In this context, it needs to be decided which information about user behavior needs to be conveyed, how this information can be tracked, and which level of realism is appropriate for the visualization. Such user representations often consist of multiple so-called *awareness cues*. In this context, the report from Oulasvirta [131] summarizes interesting background information. While the work is not specifically focused on awareness cues for XR, it provides a very concise notion of awareness cues by defining them as parts of a UI which are controlled through the activities of a remote user.

Relevant aspects about user behavior in XR settings include the user's location and head-orientation in space as well the user's hand movements. This information can be captured with different tracking technologies and then transferred into visual user representations that allow the other collaborators to perceive the user's activities.

The relevance of user embodiment, that is visualizing user behavior with body-like images,

in collaborative settings has already been emphasized many years ago by Benford et al. [10]. In more recent research, avatars [5, 133, 138, 140, 167, 57, 23, 136, 193, 144] have been proposed to provide a visual representation of the user. In this context, a variety of avatar visualizations has been considered [151]. Proposed approaches range from detailed human-like avatars to abstract avatars which only represent parts of the user's body by mapping the user's position and orientation in space to the avatar.

In 1998, Hindmarsh et al. [74] further pointed out the importance for collaborators to know which part of a scene another collaborator is able to see or currently referencing. Thus, a user's location and orientation in the scene as well as scene components which are being referenced or interacted with should ideally be visible at the same time. In this context, further visual cues which augment the basic avatar bodies have been implemented to convey user activities and aid communication among the collaborators. Sharing these cues is important for all collaboration styles, especially in distributed settings including remote assistance tasks as it allows collaborators to reference objects or locations in the shared space. For example, the user's current viewing direction can be further specified with a frame [99, 167], a ray [5, 48], or a view frustum [5, 140, 136, 57]. The corresponding data can thereby be obtained from the device orientation or eye-tracking. Furthermore, the avatar's hands can reflect the user's hand movements. Depending on the exact setting, either the movements of the user's real hands [5, 91, 92, 99, 136, 57] or the controllers [140, 167, 193, 138, 144] can be tracked. In both cases, rays can be integrated which act as laser-pointers [91, 138, 167, 57, 92, 193, 144].

2.3 Interaction in Extended Reality

XR applications can integrate different levels of interactivity. For example, in purely passive XR applications remote users can follow a meeting or task performed on site. Regarding interactive applications this dissertation distinguishes non-spatial and spatial operations. With *non-spatial operations* we refer to interactions which are performed within a defined 2D space in the XR environment. For example, non-spatial operations concern the interaction with 2D menus which are placed in the 3D space. In contrast, we use *spatial operations* to refer to operations such as the manipulation of a virtual object's position or orientation (i.e., translating or rotating objects) [68, 120, 109, 47, 87, 19, 94, 162] as well as an object's scale (i.e., changing the size of objects) [68, 120, 47, 26, 94, 162]. Spatial operations also include scene navigation. Instead of the object's position and

orientation, here the viewpoint is manipulated. While virtual environments can be very large, the physical room in which the user is located may be smaller than the virtual scene. Thus, the user cannot naturally navigate (i.e., by walking around) to all parts of the scene as the available physical space is limited. As a consequence, methods are required which allow navigating in a virtual environment while performing minimal physical movement (e.g., [180, 101]).

A very common modality for interacting with computer systems on 2D displays is offered by WIMP interfaces whereby WIMP refers to Windows, Icons, Menus, and Pointers. WIMP UIs allow selecting (i.e., pointing and confirming the choice) icons such as a folder or menu items as well as interacting with windows. This can involve opening, closing, moving, or resizing windows as well as adapting the content visible inside the window for example through scrollbars. Thus, the pointing device plays a fundamental role when interacting with WIMP UIs [42]. While WIMP offers well-established and standardized interaction styles for non-XR computer systems, a comparable interaction paradigm for XR is missing [45]. For non-spatial operations, out-of-the-box solutions often seek to map interaction paradigms from WIMP interfaces to XR. For example, when interacting with virtual 2D windows (e.g., menus), interactions can be performed in similar manners as in common 2D applications. A native interaction approach for using HHDs in XR is to apply the same touch gestures as used for non-XR applications (i.e., the user's finger acts as the WIMP-pointer). For HMDs, some manufacturers advertise the use of controllers (e.g., for the HTC VIVE Pro) or hand gestures and gaze (e.g., for the Microsoft HoloLens and Apple Vision Pro) as a pointer in WIMP-based UIs.

For spatial operations which require more degrees of freedom (DOFs), however, WIMP-based interaction can become unsuitable. As explained by Dörner et al. [45] spatial object manipulation tasks need to be divided in several subtasks if they are performed with 2D UIs and thus require higher physical and cognitive efforts. Thus, a lot of research has been conducted on the design of appropriate interaction techniques for XR [158, 73].

For HMDs, common methods to select and manipulate objects use gaze [14, 130, 97, 49, 185, 87], gestures [87, 20], controllers [130, 185], or secondary displays [145, 30, 162] which act as advanced controllers. With appropriate tracking technology, a user can select an object by looking at it or pointing at it with the hand or a controller. To manipulate the object's position or orientation, hand [87] or controller [19] movement can be mapped to the object. Controllers based on smartphones or tablets offer touch as an additional input modality [162, 94, 174]. When interacting with HHDs, objects can be manipulated

via touch [68, 120, 109], in-air gestures [16, 90], or device movement [68, 120, 109, 161]. Objects can be selected by touching them directly on the screen, by grabbing them with a gesture, or by hovering over them with the device. Manipulation can then be performed through touch-gestures performed on the display, hand gestures performed in the air, or by mapping the device's movement to the object. Further input modalities which have been explored for interaction in XR include voice [185], foot movement [121, 190], or brain-computer interfaces [156].

In this context, a relevant difference between HMDs and HHDs is that HMDs are specifically designed for spatial applications whereas HHDs were originally developed for 2D applications only. This does not only affect the availability of incorporated sensors and out-of-the-box interaction paradigms but also the user's attitude towards the device. Since HMDs have not been used for other purposes before, spatial interaction methods for HMDs do not interfere with previously learned interaction paradigms. In contrast, HHDs are ubiquitous and are being used daily for non-XR purposes. Applying these devices to XR settings poses different requirements to the interaction design. When using HHDs for non-XR purposes (e.g., writing a text message, scrolling through a document, or zooming into a picture) the device is usually held in a tilted position and almost parallel to the floor. In XR, mainly in MR, the HHD is meant to serve as a window to the XR scene. To this end, the device must be held in a higher position and rather orthogonal to the floor. In these settings, existing well-established interaction methods may not be suitable and must be unlearned by the user. For example, touch-based interaction is limited to 2D input and thus requires dividing 3D manipulation tasks in several substeps. As noted in [62] input for HHDs which is based on touch and in-air gestures requires holding the device with one hand which is prone to fatigue and occlusion.

To describe and compare the designs of previously proposed and the newly proposed spatial interaction methods in this dissertation, the process of object manipulation is divided in four steps (see Fig. 2.3). First, the object which the user wishes to manipulate needs to be selected. In this context, the *Midas Touch Problem* must be taken into account which describes the issue of unintendedly executed commands. For example, these issues can occur when actions are triggered solely by the user's gaze. On the one hand this allows easy interactions while on the other hand it also means that the user is not able to look around without triggering unintended actions [81]. In XR, this problem can occur with several input modalities. For example, a ray based on gaze, gesture, or controller input is likely to hover objects which the user does not want to select or manipulate.

To prevent the Miday Touch Problem, the selection of objects should be divided in two substeps. As illustrated in Fig. 2.3, the user should first (I) specify the desired object and then (II) confirm the selection in a second step. Only the selected object should be (III) manipulated in a third step. Thereby, the object's position, orientation, or both can be adapted. Eventually, the object needs to be (IV) released (i.e., input is decoupled from the object's movement).

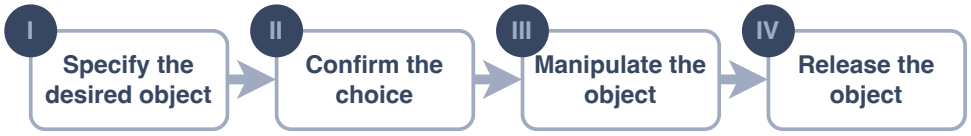


Figure 2.3: Object manipulation steps.

2.4 Further Related Aspects

This dissertation was motivated by the question of why XR technologies have seen limited adoption in practical settings, despite the considerable potential which has been assigned to them and the amount of research efforts dedicated to the topic. In this context, research on user acceptance (UA) models helped shaping the focus of this dissertation. Human Computer Interaction (HCI) research is focused on designing UIs that make it as easy as possible for humans to interact with and thus leverage the potential of different computing systems. While this includes the design of new system features that increase UA, UA is not only determined by system design and often not measurable at early stages in the development process. An important first step is therefore to understand the role of HCI research on XR-UIs in the context of UA. To this end we take into account existing models for UA.

Various models have been proposed that seek to explain UA and future utilization behavior. A model that is very popular is the technology acceptance model (TAM) which was introduced by Davis in 1986 [34]. According to TAM two core factors influence UA: the perceived ease of use and the perceived usefulness. Reflecting on his TAM model in 2006 Davis [35] associates the perceived ease of use with the absence of effort and usefulness with the expected performance increases while completing tasks with the system. Davis further points out that usefulness is rather related to the functionalities provided by a system while ease of use rather relates to the design of the interface. He further reports results from a study that indicate that to express the perceived usefulness of a system, users do not necessarily have to use the system. Instead, videos of the system features

could be used. However, this does not hold for the perceived ease of use. Here, users need to have interacted with the system to precisely report its perceived ease of use. Hence, when developing new UIs for a system, they should undergo hands-on evaluation to be able to evaluate its ease of use.

Ever since the introduction of TAM, various researchers have modified the model and presented adapted versions. In 1999, Dishaw and Strong [41] presented a model which extends TAM with task-technology fit (TTF) constructs. The TTF model was introduced in 1995 by Goodhue and Thompson [66] who assumed that UA is highly influenced by the degree to which a technology matches the demands of a task. Due to the different acceptance behavior aspects captured by the two models, Dishaw and Strong [41] suggested their combination and found that integrating TTF in TAM explains a higher amount of variance when predicting UA than either of them alone.

From these UA models we derived the following learnings:

- XR-UIs should be designed to be easy to use (i.e., reducing effort).
- Evaluating the absence of effort requires hands-on interaction.
- A thorough analysis of the capabilities of XR technologies and task characteristics is crucial before designing UIs.
- A good UI design alone can contribute to but not guarantee increases in UA.

The UIs developed and presented in this dissertation seek to reduce different kinds of effort: cognitive, temporal, and physical effort. While certain aspects of the UI design can directly affect each type of effort, there is also a relation between them. When cognitive effort decreases, users may be able to complete tasks faster such that less temporal and physical effort is required too.

With temporal effort we refer to the time needed to accomplish a task with a given UI. Regarding physical effort, we distinguish between effort caused by active movement of different body parts and constant muscle activity which can lead to fatigue. Cognitive load refers to the amount of cognitive resources that are required to perform a cognitive task [63]. Its concept is complex and relates to fundamentals from cognitive psychology which are summarized below.

In the context of the Cognitive Load Theory [165] the terms intrinsic, extraneous, and germane are used to refer to different types of cognitive load and resources. The intrinsic cognitive load is imposed by the nature of the task whereas the extraneous cognitive load is imposed by the instructional design. The cognitive resources dedicated to intrinsic cognitive load are referred to as germane resources whereas the resources dedicated to the extraneous cognitive load are referred to as extraneous resources. As cognitive resources are limited, the extraneous resources required to perform a task determine the remaining germane resources which are used for processing and learning the information presented. As such, an optimized instructional design can reduce extraneous resources and thus increase the resources which can be dedicated to the intrinsic load. To acquire new knowledge, such as for example learning how to use a UI, information must be processed in the working memory, linked and stored with knowledge from the long-term memory. In this context, element interactivity describes the amount of elements which must be processed simultaneously in the working memory. Element interactivity can be associated with both intrinsic and extraneous load and is affected by previously established knowledge. So-called schemas structure previously gained information in an abstract way in the long-term memory and can be treated as a single element in the working memory. As such, retrieving information from the long-term memory which is structured as a schema reduces element interactivity and thus the amount of cognitive resources needed to process the elements.

Based on this knowledge, we add the following two learnings:

- The design of an UI is particularly relevant when intrinsic cognitive load is high.
- Cognitive resources required to perform a task can be reduced if they can be mapped to previously established schemas.

Reducing the effort (i.e., the resources) needed to perform a task enhances efficiency and thus *Usability* of an UI which is defined by its effectiveness (i.e., the accuracy and completeness with which tasks are accomplished), efficiency (i.e., the resources used to accomplish tasks), and satisfaction (i.e., the degree to which a user's cognitive, emotional, and physical response from using a system match the needs and expectations) according to ISO 9241-11 [39]. Interaction principles which are relevant to the design and evaluation of interactive systems are presented in ISO 9241-110 [40] along with general design recommendations. The principles state that the interactive system should support the

user in completing a task, offering functions which are suited to the requirements of the task and avoiding unnecessary steps which can be executed automatically by the system (suitability for the user's task). Furthermore, the system's functionalities should be made obvious to the user (self-descriptiveness) and the system's behavior should be predictable (conformity with user expectations). Thereby, the system should minimize efforts required to learn how to use it (learnability) and allow the user to retain control of the UI (controllability). Eventually, it should support the user in avoiding and recovering from errors (user error robustness) and encourage the user to keep using it (user engagement).

Parts of this chapter have been previously published in:

V. M. Memmesheimer and A. Ebert (2022): Towards Advanced Evaluation of Collaborative XR Spaces. In C. Ardito et al. (Eds.) Sense, Feel, Design. INTERACT 2021. Lecture Notes in Computer Science, vol. 13198, pp. 443–452. Springer, Cham. doi: 10.1007/978-3-030-98388-8_40.

V. M. Memmesheimer and A. Ebert (2023): A Human-Centered Framework for Scalable Extended Reality Spaces. In J. C. Aurich, C. Garth, and B. S. Linke (Eds.) Proceedings of the 3rd Conference on Physical Modeling for Virtual Manufacturing Systems and Processes (IRTG 2023), pp. 111–128. Springer, Cham. doi: 10.1007/978-3-031-35779-4_7.

V. M. Memmesheimer, J. Löber, and A. Ebert (2024): AWARE^{S}CUES: Awareness Cues Scaling with Group Size and Extended Reality Devices. In J. Y. C. Chen and G. Fragomeni (Eds.) Virtual, Augmented and Mixed Reality. HCII 2024. Lecture Notes in Computer Science, vol. 14706, pp. 44–59. Springer, Cham. doi: 10.1007/978-3-031-61041-7_4.

Chapter 3

XR^S – Scalable Extended Reality

XR technologies allow inspecting and interacting with 3D virtual elements in physical or entirely virtual environments. Previous research has unveiled considerable potential for their application in various domains including manufacturing industries, architecture, healthcare as well as co-located and distributed collaboration in general. Even within each of these domains, XR technologies can support multiple use cases. For example, in the automotive industry, XR can be applied in (collaborative) design reviews of virtual prototypes as well as in the training of machine operation, or for providing support through remote experts. Thereby, different use cases can benefit from different XR technologies by leveraging their specific strengths.

While the field of potential applications of XR is vast, its actual implementation in real-world settings is still limited. The current use case driven development of XR applications results in solutions which are tailored to specific hardware, supported operations, and numbers of users. As such, using XR technologies for multiple use cases requires additional efforts to adapt to new systems. To exploit each technology's potential across different use cases while maintaining usability, these efforts need to be kept as low as possible. Thus, scalable UIs which stay consistent when switching between different XR technologies are required. In this chapter we address this gap through the introduction of Scalable Extended Reality (XR^S) spaces which provide scalability between different degrees of virtuality, different devices, and different numbers of users.

The chapter is structured as follows. First, an overview of promising fields for XR applications and research results from fields related to the development of XR^S is given. These concern the design of collaboration support features, the generation of consistent and accessible scenes, and the implementation of appropriate interaction techniques. Ad-

dressing the limited scalability of existing XR solutions, we present our vision of highly scalable XR environments named XR^S in the second part of this chapter. Based on the literature reviewed and the presented XR^S concept, barriers to the realization of XR^S are identified and translated into a research agenda. To set the stage for XR^S and as the basis for this dissertation, the third part of this chapter presents a framework for XR^S spaces. To do so, abstract use cases exploiting XR's key benefits are defined. Based on them, requirements are formulated and translated into the framework. Eventually, two exemplary walkthroughs are given to explain how the framework can be applied to specific use cases which can be described by (combinations of) the abstract use cases.

3.1 Related Research and Aspects

This section provides an overview of fields in which the application of XR technologies is deemed promising and summarizes research which is related to XR^S such as collaboration support features, scene generation, and interaction techniques.

3.1.1 XR Applications

Research on potential applications of XR technologies has revealed promising use cases in various fields. They have been assigned considerable potential to support design and engineering reviews in many domains as XR technologies allow to intuitively inspect and interact with spatial content. For example, Kaluza et al. [86] integrated visual analytics methods to an MR application to support decision-making in automotive life cycle engineering, Gong et al. [65] developed a multi-user VR application for cooperation among globally distributed users during an automotive design review task, and Wolfartsberger [188] developed and evaluated a VR application for design review of power units.

Another potential use case which is relevant across many industries is factory layout planning. In this context, the VR system from Gong et al. [64] seeks to facilitate the modeling process and to improve decision-making through more accessible visual representations. XR technologies can be applied in a similar way for interior design such as for example presented by Vazquez et al. [177]. Their MR tool integrates scale-accurate furniture as virtual augmentations.

On top of that, XR can be applied in the context of construction, for example for perform-

ing safety training in VR such as presented by Wu et al. [189] or for supporting monitoring and documentation tasks with virtual augmentations such as presented by Zollmann et al. [197].

The work of Pirker [135] provides an overview of potential applications of VR technologies in aerospace. These include training in simulations, teleoperating remote machines, testing, design reviews, collaboration, and remote assistance.

Apart from this, further interesting fields of application for XR can be found in healthcare. Here, the literature review from Sadeghi et al. [148] presents interesting applications of XR in the context of cardiothoracic surgery, including surgical planning, training in virtual simulators, and intraoperative guidance. For example, information which are visually augmenting the surgeon's field of view can be scaled and placed according to the surgeon's personal preferences [3]. In a similar way, maintenance tasks can benefit from XR by integrating relevant information directly into the worker's field of view [157].

Such visual instructions can either be provided automatically by the system or by a remote collaborator. XR technologies cannot only support co-located collaborators but are also predestined for sharing spatial information among distributed collaborators. For example, the system presented by Bai et al. [5] allows sharing a local working space with a remote collaborator who can provide support through visual cues that augment the local worker's field of view.

3.1.2 Collaboration Support Features

XR-supported collaboration can be very different from face-to-face collaboration and natural communication. Addressing these barriers, previous research proposed several solutions for collaboration in co-located [184, 68, 193] and distributed [5, 91, 92, 99, 37, 133, 138, 140, 167, 193, 176, 23, 136, 170, 144] scenarios, using HHDs [184, 68, 176], HMDs [5, 91, 92, 99, 37, 138, 140, 136, 170, 167, 23, 193, 144], or both [57, 133, 107, 96, 56, 175, 44].

3.1.2.1 Avatars

As described in Chapter 2.2, collaborative XR systems commonly use avatars to represent user behavior. In this context, previous research considered different approaches for tracking and visualization. Thereby, generating full-body avatars requires advanced methods.

For example, Yu et al. [192] tracked HMD users based on their shadow to create the corresponding avatar in real time. Further approaches use inverse kinematics to generate avatars of HMD users [138, 193, 43, 144]. Inverse kinematics algorithms allow computing avatar postures that cause the avatar's head and hands to attain a predefined (e.g., obtained from tracking) position and orientation [45]. Inverse kinematics have also been applied for generating avatars for HHD users [122, 105, 2]. Here, tracking the user's head orientation is not as straightforward as for HMD users. Thus, less data is available which describes the user's actual pose which makes the generation of appropriate user representation more challenging. Another approach for tracking HHD user poses in co-located settings was presented by Ahuja et al. [1] who estimated body postures by combining the views of multiple co-located users' HHD cameras.

In a second step, the tracked user's pose then needs to be transferred to an appropriate visualization. In this context, different degrees of realism have been proposed ranging from human-like avatars [138, 140, 193] to more abstract user representations [133, 57, 5, 167] which include virtual visualizations of the respective XR device or generic objects such as a sphere with a view frustum.

Researching the ideal design of avatars also requires overcoming the so-called *uncanny valley* – a term which refers to Mori's work on the perception of humanoid robots [119] and describes the phenomenon that an almost but not perfectly realistic visual representation of a human is perceived as uncanny. More specifically, affinity for the visual representation does not monotonically increase with but drops at a certain level of human-likeness.

Apart from this, it has been pointed out that life-size avatars may not be suitable in all situations as they can occupy a large part of the field of view. Thus, the usage of miniature avatars has been explored. Addressing this issue, Piumsomboon et al. [138] propose a miniature avatar which represents an off-site collaborator by mimicking the respective gaze direction and gestures. More specifically, a miniature of the life-size avatar representing the off-site collaborator is created. It is equipped with a pointer and represented the body movements of the corresponding remote collaborator. Thereby, the location of the miniature avatar was updated according to the current field of view of the local worker while a ring at the miniature's feet indicates the off-site user's and life-size avatar's location.

3.1.2.2 Visualizing Viewing Directions

To make a user's viewing direction visible for other collaborators, avatars have been augmented with additional cues. Frequently applied techniques for capturing the user's viewing direction are head tracking [136, 5] and eye tracking [48, 136, 5]. In addition, research is being conducted on the comparison of different tracking methods [183, 181].

Previous research has considered different visualizations for conveying a user's viewing direction. For example, Bai et al. [5] used a generic sphere to represent an absent collaborator's head. The basic orientation of the user was indicated through a view frustum. Additionally, a ray emerging from the sphere indicated the user's gaze direction. View frustums were also used by García-Pereira et al. [57] and combined with a visual representation of the respective device to represent an HMD, HHD, and desktop user. Instead of rendering frustums, Bovo et al. [17] visualize a user's viewing direction through the intersection of a user's cone of vision on 2D surfaces.

Lee et al. [99] found that augmenting a collaborator's view with a rectangle to represent the other collaborator's current viewing direction, failed when using HMDs whose fields of view differed in size. As a solution, the size of the rectangles was reduced according to the size of the output device's field of view. Similar issues were reported in [136].

To indicate the remote collaborator's viewing direction, the miniature avatar's face from Piumsomboon et al. [138] was always directed in the viewing direction of the user who was represented by the avatar. In another work from Piumsomboon et al. [140] a miniature avatar was augmented with a view frustum whereby the view frustum included a video stream of the respective user's current view. Similarly, Teo et al. [167] combined a live video with a more abstract avatar representation.

3.1.2.3 Visualizing Hand Movements

Apart from a user's position, orientation, and viewing direction, the user's hand movements are commonly shared among collaborators. Depending on the specific setup, previous work sometimes only captured the off-site collaborator's hands, as the hands of the on-site collaborator are often captured within the virtual replica of the MR space. For tracking the off-site user's hands, previous research considered directly tracking the hand through a sensor attached to the HMD [5, 91, 92, 99, 136, 57] as well as tracking a controller held in the hand [140, 167, 193, 138, 144].

The tracked data is then transferred to animate an avatar's hands [5, 91, 92, 193, 136] and to generate a laser pointer [91, 92, 138, 167, 193, 96, 57, 144]. In some approaches [193, 136], the visualization of hand gestures from remote collaborators was limited to predefined gestures such that if one of the predefined gestures was detected, the avatar's hand was adapted accordingly. Concerning the pointer visualization, in some approaches the pointer is always on [193, 144] whereas in others it can be activated with specific user input [91, 92, 57, 138, 167].

For example, Piumsomboon et al. [138] activate the ray when the trigger button of the controller is pressed. On the contrary, Kim et al. [92, 91] activate the ray upon the detection of a pointing gesture. A similar approach was provided by García-Pereira et al. [57] for HMD users. For HHD users, their system includes a virtual tablet and a virtual hand which appears along with a ray when touch input is registered. Similar rays for HHDs were also proposed in [107, 176]. In the system by Kostov and Wolfartsberger [96] the HMD user's pointer emerged from the controller whereas for the HHD it emerged from the camera's center. In the collaborative HHD setting from Grandi et al. [68] rays connecting the HHD and the currently selected object are integrated into the MR scene. In addition, they integrated icons that indicate the specific manipulation a user is currently performing with the selected object. Thereby, different colors are used to distinguish between users.

The approach from Ibayashi et al. [79] allows collaboration among an HMD user and a tabletop display user. Tabletop users can point at specific locations on the tabletop display which shows the virtual space from the top. The HMD user will then see large virtual hands pointing at the respective location. At the same time, the HMD user could point at certain locations by applying touch on an HHD which is mounted in front of the HMD. The tabletop users will then see an avatar pointing in this direction.

Further approaches [91, 92, 193] allow the off-site collaborator to draw virtual sketches into the shared environment. An alternative to sketching for remote instruction was presented by Tian et al. [170]. The method involved virtual replicas of physical objects which are manipulated by the remote collaborator to provide instructions to a local user.

3.1.2.4 Visualizing Out-Of-View Elements

Since collaborators may have different viewpoints in the shared space, visualizing elements which are outside a collaborator's field of view motivates further research.

In the approach presented by Lee et al. [99] a virtual hand is augmented with a glowing effect such that it is easier for the other collaborator to find the virtual hand if it is outside the current field of view. This design relates to Halo [8], a technique for visualizing off-screen elements which was originally developed for 2D applications. A similar glowing effect was added to the miniature avatar proposed in [138].

Another approach for attracting a remote collaborator's attention was proposed in [140] where the off-site user was given the option to attract the on-site collaborator's attention through a virtually burning torch. Furthermore, virtual arrows pointing to the location of remote collaborators or their viewing direction have been integrated to XR environments [5, 99].

3.1.2.5 Access Control

Apart from user representations, collaborative XR settings require appropriate solutions for handling access control. Since many of the proposed collaborative XR systems focus on remote instruction scenarios where the off-site user solely guides the on-site user with visual cues, object manipulation is often only possible by the on-site user and thus access control was not relevant in these contexts.

Research papers which involve collaborative object manipulation considered different approaches for this. For example, Grandi et al. [68, 67] allowed co-located collaborators to simultaneously manipulate an object by multiplying each transformation matrix with the one of the virtual object. Wieland et al. [186] compared three different ways of merging interactions of HHD collaborators: separating translation and rotation among collaborators, allowing the users to only perform the same manipulation operation simultaneously, and a hybrid approach allowing both users to perform the same or different manipulations simultaneously. Based on the results of their comparative study, they recommend the separation approach in time-critical scenarios and hybrid if the main goal is to enhance user experience. On the contrary, Wells and Houben [184] allowed object manipulation only by one of the co-located collaborators at a time. During manipulation, the object was locked for the other users as indicated by a colored border around the screen of the HHD. In the approach of Pereira et al. [133] a master client was supposed to give ownership to the other clients.

3.1.3 Scene Generation

For effective collaboration among distributed collaborators, off-site users should be provided with a virtual reconstruction of the on-site environment. This virtual reconstruction must remain consistent with the on-site environment, meaning that changes in the physical environment need to be adapted in the virtual scene as well. Additionally, the virtual reconstruction must be accessible, allowing off-site collaborators to reference and interact with scene components. Thus, to ensure that both on-site and off-site users have access to the same information and options for interaction, virtually reconstructed scenes need to be segmented semantically and adapted dynamically.

3.1.3.1 Virtual Reconstruction

Previous research used different approaches and technologies to capture the on-site environment such as 360-degree cameras providing pictures [167] or videos [99, 140, 167], light fields constructed out of pictures taken with a smartphone [118], multiple RGB-D [103] cameras, as well as built-in spatial mapping [167, 166] to generate a mesh of the environment. While in some approaches the on-site environment was captured prior to the actual collaboration [118, 166], other approaches [99, 140, 103] allowed the virtual scene to be updated in line with changes on-site. The virtual reconstructions could be accessed via VR-HMDs [99, 133, 140, 167, 103, 166] or HHDs [118].

Ideally, the virtual replication of the physical scenes should provide high visual quality, live updates, viewpoint independence, and bidirectional manipulation. Often, however, trade-offs among visual quality and further quality criteria exist: For example, 360-degree cameras deliver high visual quality but also restrict the off-site collaborator's viewpoint according to the camera's position. In contrast to 360-degree pictures which are static, 360-degree videos can provide the off-site collaborator with a dynamic representation of the physical scene. A common approach to achieve this is to mount the 360-degree camera on the head of an user who is located on site and to then provide the off-site collaborator with the corresponding video stream [99, 140]. As noted by Lee et al. [99], the orientation of the 360-degree video will here, however, always depend on the on-site collaborator's head motion. To provide the remote collaborator with an independent view of the scene, they tracked the head orientation of the on-site collaborator and adjusted the off-site collaborator's view accordingly.

Another problem which occurs when users have different viewpoints, concerns the placement of 2D annotations in the 3D scene [118, 58, 59, 60, 128, 129, 102]. Simple virtual annotations which are not semantically anchored in space will become useless when the collaborators have different viewpoints (i.e., when the collaborator seeing the annotation has a different viewpoint than the one from which the annotation was added).

Apart from this, virtual scene reconstructions using 360-degree cameras usually prevent bidirectional manipulation. Thus, off-site collaborators who are provided with the video stream can see the entire scene but not manipulate scene components. Teo et al. [167, 168] allowed off-site collaborators to choose between a 360-degree video stream for a dynamic, high-quality visualizations and a static 3D mesh textured with a 360-degree picture to explore the space independently of the on-site collaborator. In [169], they developed this approach further by allowing off-site collaborators to retexture the static mesh with different 360-degree pictures. However, they found that this approach is likely to produce holes in the virtual scene at spots where objects in the picture occlude each other.

Using depth cameras to obtain geometric reconstructions of physical scenes can enhance the off-site collaborator's spatial awareness of the scene and allow exploring the space independently of the on-site collaborator's location. In some approaches sensors incorporated in the Microsoft HoloLens captured a static reconstruction of the scene prior to the actual collaboration [167, 166, 168, 169]. However, the integration of live updates is challenging, as only those parts in space can be updated which are captured by the depth camera.

Lindlbauer and Wilson [103] presented an approach for providing live updates independently of the user's position and orientation. This, however, requires a more complex setup involving several RGB-D cameras which provide a geometric live reconstruction of a physical room. Their approach allows the user to perform detailed interaction with real-world objects. To this end, interactions were performed on a voxel grid (i.e., a volumetric representation of the reconstructed space where each voxel held information on possible manipulations). However, Mohr et al. [118], note that the visual quality of such approaches may be impaired by shiny and transparent surfaces. These surfaces are common in industrial environments – a field in which the application of XR technologies is deemed particularly promising. Addressing this issue, they proposed to provide the off-site collaborators with a light field: a high-quality, yet static visualization of the workspace. Therefore, the on-site collaborator used an HHD to take pictures of the scene which were stored along with the HHD's position and orientation. The off-site collaborator could

then add 3D annotations to the generated light field.

While increased visual quality and accessibility is expected to enhance collaboration, it will also increase the amount of data that needs to be processed. This can cause latency which could impede communication among collaborators. Hence, another relevant research topic concerns the optimization of data processing. In this context, Stotko et al. [160] presented a framework for sharing reconstructed static scenes based on RGB-D images with multiple clients in real time. Moreover, the amount of visual data which needs to be processed may be minimized by the incorporation of digital twins. Physical objects whose appearance can be described by a digital twin would not have to be tracked visually. Instead, the virtual appearance of the object could be rendered according to its digital twin's current state. This might also allow the manipulation of real components through the off-site collaborator. For instance, Jeršov and Tepljakov [82] presented a system which allowed altering the water levels of a physical multi-tank system by manipulating the system's digital twin in a virtual environment.

3.1.3.2 Semantic Segmentation

To support bidirectional interaction in XR^S, off-site collaborators should further be able to access and reference objects in the joint space. A semantic segmentation of the reconstructed space would allow storing additional information as well as predefined manipulation behavior in semantically segmented objects and is hence expected to accelerate object selection and to facilitate interaction. Schütt et al. [155] presented an approach that allows to semantically segment physical spaces while using an MR application. Thereby, a mesh captured by the Microsoft HoloLens is semantically annotated and back-projected onto the physical scene such that real-world objects could be highlighted, offering object-specific manipulation options upon selection. Automated annotation of 3D scenes could further benefit from deep learning. Dai et al. [33] note that in this context labeled training data is often missing. Addressing this issue, they present a system that allows capturing 3D scenes via RGB-D scans and annotating the corresponding 3D meshes. As such, they were able to collect a large data set of labeled 3D scenes. Huang et al. [75] took use of this data set for semantically segmenting a 3D space described by supervoxels in real time.

3.1.4 Interaction Techniques

This section provides an overview of the variety of interaction methods explored in previous research. In line with the steps of manipulation described in Chapter 2.3 we first summarize selection methods and then manipulation methods. While some of the respective papers focus solely on selection, others primarily focus on manipulation, and some address both.

3.1.4.1 Object Selection

Regarding the selection of virtual items, previous research has considered different approaches for HHDs and HMDs.

For HHDs objects can be selected similar as for non-XR applications by touching the object directly on the screen [186, 120, 90]. In this case, the two steps of selection (i.e., specifying the object and confirming the choice) are performed together. Various touch-based methods for selecting small potentially occluded objects seen through MR-HHDs have been investigated by Yin et al. [191].

Apart from touch, in-air gestures were developed which allow MR-HHD users to point at objects [16]. Less commonly, tangible selection was considered. For example, Wacker et al. [182] provided a tangible input modality for MR-HHDs. A 3D-printed pen with visual markers could be used to select virtual objects. Thereby, small spheres or rays represented the pen's position and buttons could be used for selection confirmation.

However, these selection paradigms require holding the HHD with one hand which is deemed unfavorable [62]. Alternatively, objects can be selected with a screen-centered cursor [15]. Such an approach would allow holding the HHD with both hands while it also requires a second modality (e.g., pressing a button) for confirming the choice.

A similar selection paradigm is frequently considered for HMDs. Here, users can either specify objects by controlling a cursor through head or eye movements. These methods have been compared in multiple studies [14, 97, 142] which delivered mixed results. Theoretically, eye-tracking technology can capture a user's viewing direction more precisely whereas the head orientation is only a proxy for the user's viewing direction. However, Jacob and Stellmach [80] note that unconscious eye movements can impede interaction based on eye movements. Furthermore, they note that a user's eyes are not solely used for

specific input but also for perceiving the environment. These eye movements can interfere with those for selecting objects. The same issue can also occur for selection based on head movement and refers to the midas touch problem. Thus, an appropriate method is needed for confirming a potentially selected object.

For this, previous work considered different modalities such as dwell time (i.e., selection is automatically confirmed after the cursor hit the target for a defined time) [14, 49, 130, 77], hitting an additional confirmation flag with gaze [117], using gestures (e.g., performing a circle [49] or a tap gesture [97, 87]), performing a click with a button on the HMD [49] or an external device [49, 97, 130, 30], as well as voice [49, 185]. Piumsomboon et al. [137] point out the time constraints in dwell-based confirmation. Seeking to prevent waiting times if dwell times are too long and unintended selection if dwell times are too short, they present more advanced confirmation options which are based on natural eye movements. On top of that, Kytö et al. [97] allow refining the selection based on eye or head movement through different options before confirming the selection. For this second step the placement of the selection cursor can be further adjusted by rotating a controller, performing in-air gestures, and moving the head.

Apart from gaze-based selection, hand tracking was employed for selecting objects. Previous work has attached an additional sensor to the HMD which captures the user's hand. This approach was also followed by Kang et al. [87] who allow users to directly grab objects. While hand tracking through sensors incorporated in the device reduces the setup complexity and enhances mobility, users need to make sure that the hand is always placed in a trackable position. Previously, an alternative approach for tracking the user's hand and arm movements was to place markers to the user's arm and body which are tracked by multiple cameras [154]. Seeking to reduce fatigue, which is likely to arise if the hand needs to be held up high to be captured by the device sensors, Brasier et al. [20] used the Leap Motion controller to create a virtual plane at the user's waist, thigh, or wrist. Users could point at targets by moving the hand to the respective position in the virtual plane and keeping the cursor at the point of interest confirmed the choice. In this way, a more comfortable input position for small-scale hand movements to control a cursor in MR was offered.

Instead of pointing with the hands this can also be done with external devices. In this context both, conventional controllers [130, 185] as well as smart devices [145, 30, 89, 196, 162, 143] have been considered. A common approach here is to transform the smart device into a laser pointer [145, 30, 196]. At the same time, the touch interface of the smart

device serves as an additional input modality for further interactions, e.g., confirming the selection.

In this context, Chen et al. [30] considered both smartphones and smartwatches as laser pointers. The cursor could thereby either be controlled only with the smart device (i.e., the cursor could leave the field of view) or with a hybrid approach (i.e., the cursor always stayed within the field of view such that large-scale movements could be performed by moving the head and small-scale movements inside the current field of view could be performed with the smart device). Selections could be confirmed either by tapping on the touch interface or by wrist rotation (smartwatch). Furthermore, they considered using the touch surface of the smart device as a trackpad to control the cursor. Here, however, the smartwatch display turned out to be too small. Another approach involving a tablet for controlling a cursor in an area specified by the user's head orientation was presented by Biener et al. [12]. Apart from this, Lee et al. [100] proposed different (force) touch gestures to navigate a cursor to the starting point of a text section. Selections could be confirmed through circular touch gestures or by selecting the end point. A similar approach to [30] was followed by Le et al. [98] who mapped the user's current focus area (determined by head movement) on a large display to a tablet. The hand interacting with the tablet was captured by a camera and mirrored to the large display seen through the HMD. Thus, the user could directly select objects by touching the respective part of the display. Kari and Holz [89] allow users to move virtual hands in 3D by combining input based on device movement and touch gestures. Furthermore, grabbing metaphors have been implemented which allow picking up virtual objects by touching them with the device [143, 162] as well as touching the object directly on the screen.

Apart from this, previous work has considered using the user's feet to select items. Xu et al. [190] propose a motion-based approach for selection. Thus, to select a menu item the user has to step in the direction representing the menu item. Other interaction designs which could reduce arm fatigue during selection rely on input via gaze and feet. For example, Müller et al. [121], proposed an interaction for users wearing an MR-HMD via tapping on the floor. The foot is optically tracked and can be used to select items which are either projected on the floor or in mid-air.

Another method for spatial selection was presented by Besançon et al. [11]. Therefore, a shape drawn on the touch interface could be used to brush through a 3D data space (i.e., by physically moving the HHD). Furthermore, tangible interaction modalities allow selecting (i.e., adjusting) the data seen through an HMD by interacting with sliders on

the custom tangible device [31] or by moving the tangible to the desired item [47].

3.1.4.2 Object Manipulation

The selection of the object can, but does not necessarily have to, be followed by its manipulation. Therefore, suitable modalities are required that enable adapting the object's position, orientation, or size – each with 3 degrees of freedom (DOFs).

While touch input has proven to be a well-suited interaction modality for 2D HHD applications, transferring well-established (multi) touch gestures to spatial object manipulation is not straightforward. The difficulty of mapping 2D touch-input to 6 DOF manipulations (i.e., translation and rotation) was addressed by Fuvattanasilp et al. [55] who proposed an approach which reduces the number of DOFs which can be manipulated at once. Their technique for placing virtual objects involves first defining an initial 2D position using touch. The depth can then be adjusted on a ray originating from this point. The orientation of the object being manipulated is limited to rotations around its gravity vector. Mossel et al. [120] implemented touch-based interaction which combines single touch input with the HHD's current pose. In this way, 2D translations on the screen are backprojected to translate and rotate the object. Thereby, mode switching is enabled through buttons. Similar implementations for translations in the xy-plane were used in [68, 109]. Grandi et al. [68] enable z-axis translations by one-touch tapping and sliding, x- and y-axis rotations via two-touch sliding, and z-axis rotations via two-touch rotations. Marzo et al. [109] used the distance and angle between two touch points to handle z-axis translations and rotations as well as their middle point to control an arcball rotation.

Apart from touch-based interaction, hand gestures were considered for manipulating objects. Botev et al. [16] presented an approach using one-handed gestures to grab and move virtual objects seen through MR-HHDs. Kim and Lee [90] presented a hybrid approach in which touch is only used for object selection and object manipulation is performed via hand gestures. To this end, an external sensor which detects the hand gestures was attached to the HHD. The AR-Pen presented by Wacker et al. [182] could be used for moving virtual objects seen through a MR-HHD. After object selection, the object could be dragged and dropped by pressing and releasing buttons on the physical pen.

As an alternative to these one-handed, and thus fatigue and occlusion prone [62], input modalities it has been proposed to map the movements of an HHD (i.e., changes in its position and orientation) to virtual objects [68, 120, 109, 161]. This device-based

interaction allows holding the HHD with both hands.

For manipulating objects seen through HMDs, common interaction modalities involve in-air gestures and controllers. Thus, the user's hand movements can be mapped to the object while a specific gesture is performed [87, 185, 26] or the controller movement can be mapped to an object while a specific button is pressed [185, 19]. Such interaction paradigms are common out-of-the box solutions for HMDs like the Microsoft HoloLens and HTC VIVE Pro.

Apart from conventional controllers, interaction for HMDs was enabled through the integration of smart devices which act as advanced controllers. Here objects can be manipulated through touch gestures (i.e., the smart device acts as a touchpad) [145, 94], through device movements [104] (similar to conventional controllers) or both [174].

Furthermore, Kang et al. [87] implement object manipulation through a worlds in miniature (WIM) – an interaction metaphor which was introduced in 1995 [159]. WIM allows users to adapt (i.e., manipulate objects in) a virtual environment through a second viewport, i.e., a miniature representation of the virtual environment. In this way, WIM seeks to make it easier for users to interact with distant objects and provide a better overview of the environment (i.e., which is not limited to the current field of view). When an object is manipulated in WIM its counterpart in the life-size representation is also adapted accordingly. In Kang et al.'s [87] approach users could manipulate objects by directly grabbing and moving their miniature.

Further approaches considered voice for object manipulation. For example, the approach from Whitlock et al. [185] translates objects in a 2D plane and rotates them to the left/right at a constant speed based on detected voice input. During tangible interaction virtual content seen through an HMD can be manipulated by manipulating the tangible object such as a physical sphere in [47] or a cube in [19].

3.1.4.3 Further Operations

Apart from translating and rotating objects, methods for further interaction have been developed.

Regarding adaptations of the object's size, Kiss et al. [93] evaluated different zooming techniques for MR-HMDs. For zooming in and out, users could either move one pinched hand or an external controller along an imaginary axis, move two pinched hands towards

or away from each other, or use voice commands (i.e., the keywords *smaller* or *bigger*). For touch-based interaction on HHDs, the popular multi-touch zoom gesture was implemented allowing users to scale objects on the HHD's screen by pinching and spreading two fingers [68, 120]. This gesture was also implemented for scaling objects seen through HMDs through HHD-based controllers [94, 162]. A similar approach was also applied for in-air gestures in [26] where users can scale objects by moving their pinched hands towards or away from each other. Furthermore, Grandi et al. [68] allow users to scale objects by performing a sliding gesture while objects are moved based on the HHD's movements. In [47] the tangible sphere can be rotated to scale an object seen through an HMD.

Another research topic which is particularly relevant in VR concerns the navigation inside a scene. In this context, previous research has explored different modalities. For example, von Willich et al. [180] considered different input techniques based on the position, pressure, and orientation of a user's feet for locomotion. Liang et al. [101] proposed navigation in a scene seen through an HMD based on touch input and device movement of an HHD. Another approach involving touch-gestures for scene navigation while wearing an HMD was presented in [195]. Eventually, Satriadi et al. [150] propose enhanced in-air gestures which seek to reduce fatigue while navigating on large maps.

3.2 The Vision: A Pathway to XR^{S}

Despite the potential assigned to XR technologies and the ongoing research, the actual implementation of XR in practical settings is still limited. Since different XR technologies offer different strengths from which users can benefit in different use cases, the optimal utilization of XR's potential requires seamless switching between the technologies.

Addressing this requirement, the following section introduces the concept of Scalable Extended Reality (XR^{S}) spaces which provide multidimensional scalability enhancements. In addition to the introduction of the general concept, scalability limitations of existing solutions are identified, and a research agenda is established based on a compilation of the related research summarized in the previous section.

3.2.1 Introducing XR^{S}

The current use case driven development of XR applications results in solutions that are limited with respect to hardware, tasks, and numbers of users. This lack of scalability forces users to re-adapt to systems when using XR for different tasks. Aiming to reduce the temporal and cognitive efforts required when switching between different technologies and to foster XR's application in practical settings, we introduce the concept of Scalable Extended Reality (XR^{S}) spaces which scale across different devices and degrees of virtuality and can be entered by multiple potentially distributed users (see Fig. 3.1).

Scalable Extended Reality (XR^{S}) spaces provide scalability across different

- *degrees of virtuality* (i.e., from entirely virtual environments to physical environments that are augmented with many or few virtual components),
- *devices* (i.e., they are accessible via both HHDs and HMDs), and
- *numbers of users* (i.e., from single users to multiple collaborators who may be located at different sites).

In this context, higher *scalability* is linked to more seamless switching between different UIs when accessing an XR^{S} space.

For example, XR-supported product development processes could benefit from XR^{S} spaces that scale from completely virtual prototypes to prototypes that combine single physically existing parts with virtual augmentations of the missing parts to almost physically complete prototypes that are augmented with single virtual elements (i.e., the degree of virtuality decreases as the product evolves). Similarly, degrees of virtuality could decrease according to learning progress during XR-supported training or according to progress at construction sites. Furthermore, teams of co-located collaborators operating in MR could be joined by off-site collaborators who enter a virtual replica of the scene. Thereby, each user could be provided with an HHD or HMD depending on the specific use case and preferences. In XR^{S} spaces, scalable UIs that increase memorability would allow users to intuitively switch between HMDs and HHDs as well as between different degrees of virtuality while keeping their focus on the actual task.

Depending on the target platform, the development of XR applications involves the combi-

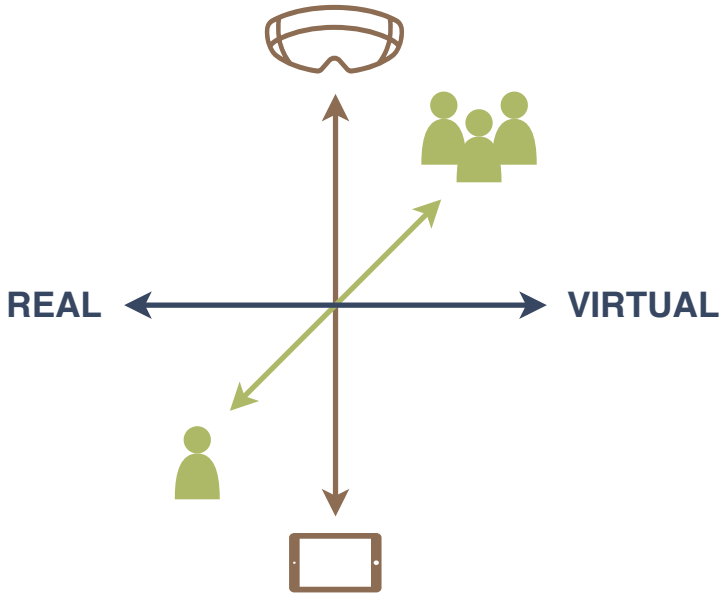


Figure 3.1: Conceptual description of XR^S spaces which provide scalability across varying degrees of virtuality, devices, and numbers of users.

nation of several tools. Seeking to reduce this fragmentation and facilitate cross-platform development, several frameworks and packages have been proposed and integrated into GameEngines like Unity. For example, the AR Foundation package in Unity provides interfaces to the native SDK of the target platform such as Apple ARKit, Google ARCore, or OpenXR which allows building to any XR device for which an OpenXR runtime exists. As such AR Foundation seeks to facilitate accessing features such as for example device tracking or plane detection. Furthermore, the XR Interaction Toolkit connects to native SDKs and abstracts interactions into components which facilitate the implementation of interactive experiences across platforms. Its AR features can be used by combining XR Interaction Toolkit with the AR Foundation package.

While these solutions address scalability issues from the developer's perspective, little attention is being paid to enhance scalability from a user's perspective. Commercial solutions such as Microsoft Mesh, Meta Horizon, or NVIDIA Omniverse seek to reduce entry barriers to (collaborative) XR. However, the landscape of collaborative XR solutions remains fragmented with a part of existing solutions having a strong focus on social experiences whereas others focus on more complex use cases (e.g., integrating digital twins). Furthermore, even though the corresponding immersive environments can often be accessed with different XR technologies, they still rely on device specific interaction modalities. Apart from this, research on transitional interfaces [69, 83] is exploring how users can be guided when transitioning between different interactive environments (e.g.,

MR and VR) while preserving presence of a coherent experience. However, consensus on the design of adequate user interfaces including collaboration support features and interaction techniques is still missing.

While many research topics relevant to the realization for XR^S have been investigated in the past, these topics have rather been explored independently of each other and little research has considered the integration of results from these research fields to enhance scalability.

For example, Pereira et al. [133] propose a solution for collaborative interior design. Users can access the presented application with VR-HMDs or MR-HHDs. Thereby, the users are represented by different avatars depending on the device used. All users can highlight parts of the scene and interact with virtual elements. To this end, device-specific interaction paradigms can be used (i.e., motion controllers for VR-HMDs and touch input for MR-HHDs). Interaction with virtual replicas of physical objects is not provided.

Furthermore, a proof-of-concept application for collaboratively training engine construction was presented by Kostov and Wolfartsberger [96]. Thereby, access to the application is possible via a VR-HMD, MR-HMD, MR-HHD, and a desktop PC. The virtual replication of physical scenes is not addressed in their approach. In contrast to [133], they point out the challenges of device-specific input paradigms. To facilitate both the implementation and the usage of interaction paradigms, they implement the same button-based interface for all devices. Users can then click the buttons with the device-specific input modality to manipulate objects according to predefined increments. However, this approach limits flexibility and controllability.

The system presented by García-Pereira et al. [57] provides the VR-HMD user with a static virtual reconstruction of the previously scanned on-site environment. Further access points are offered through a desktop PC and an HHD which can display both the VR and an MR scene. Device-specific input modalities are provided for pointing at certain objects or drawing annotations: An external sensor capturing hand gestures was connected to the HMD, the HHD's interface was based on touch, and the desktop application responded to mouse clicks. Furthermore, different avatars were generated depending on the XR device.

Zaman et al. [193] use 360-degree video streaming to enable remote collaboration. By accessing this virtual replica, VR-HMD users can follow the activities of MR-HMD users who are located on site. Thereby, VR-HMD users are represented with personalized avatars whereas the local users appear in the 360-degree video stream. A picture-in-picture

window displays a user's current perspective to the other collaborators. Communication is enabled through spatial audio and visual cues. Using their controller, VR-HMD users can draw annotations, point at targets, and perform hand gestures.

3.2.2 Defining a Research Agenda

Previous research has primarily focused on enhancing collaboration support features, scene generation, and interaction techniques. However, little attention has been given to the integration of research outcomes to enhance scalability such as proposed in XR^{S}. Addressing this gap, we compile the results from previous research and summarize research topics related to the realization of XR^{S} spaces as well as expected contributions among them in a research agenda (see Fig. 3.2).

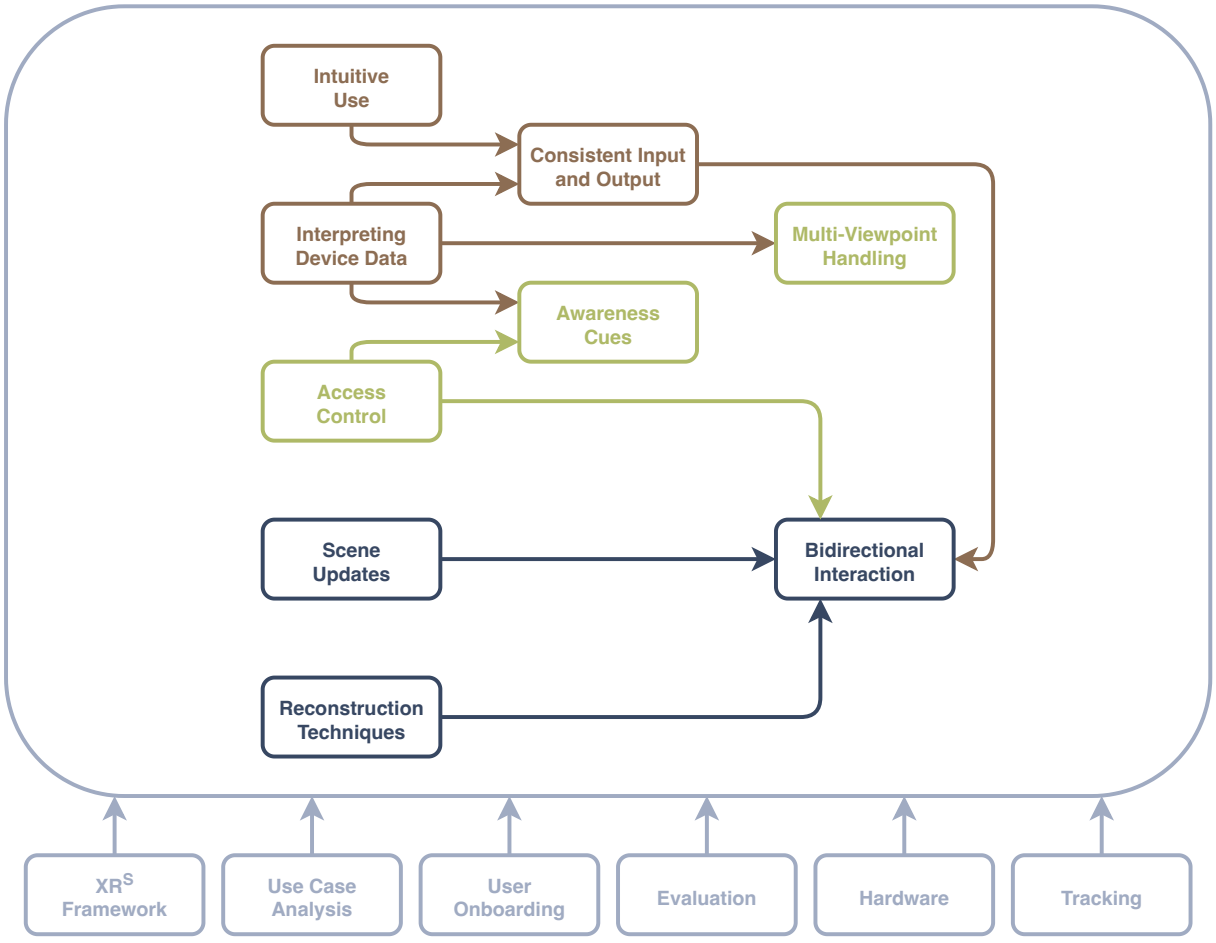


Figure 3.2: The agenda provides an overview of the research topics which are relevant to the realization of XR^{S} spaces: General research topics as well as research to enhance scalability across varying degrees of virtuality, devices, and numbers of users. The arrows depict expected contributions among the outcomes of the respective research.

The agenda is structured into research topics concerning the general implementation of XR^S as well as research on scalability enhancements concerning different degrees of virtuality, different XR devices, and different numbers of potentially distributed users.

3.2.2.1 General Research Topics

XR^S Framework. The implementation of multi-dimensional scalability enhancements such as proposed in XR^S and the incorporation of different XR technologies in a joint XR^S space requires scalability enhancements from both the developer's and user's perspective. Thus, a conceptual framework which formalizes user roles, interactions with real and virtual components, as well as data tracking and sharing among the entities is needed as a foundation for implementing XR^S in practical settings.

Hardware. Apart from the lack of scalable UIs, the design of XR hardware also heavily influences its practical adoption. Hence, upgraded hardware is needed which enhances ergonomic aspects (i.e., reducing weight), screen size, and quality.

Tracking. Precise tracking is relevant to several XR-related research areas. In collaborative applications, user representations of other collaborators can only be as good as the data provided by the tracking system. Apart from this, XR^S requires precise tracking of real scene components to provide off-site users with appropriate virtual replicas. In previous research, hardware for tracking human body behavior as well as the scene itself was often attached to the users. For example, hand-movements were tracked by sensors attached to the HMD. In this way, tracking depends heavily on the user's hand and head movements. Similarly, tracking real scene components with cameras attached to an on-site user is affected by their head movements and position. In XR^S the virtual replication of both collaborators and real scene components should be independent of other user behavior and off-site users should be able to explore the joint space independently.

Evaluation. The evaluation of XR applications and their features often leaves a large gap to real-world use cases in terms of the abstraction of the tasks, the duration of the study, and the involved participants. Tasks for which the features are implemented and evaluated are often simplified and very different from their potential application. Due to limited resources, long-term studies are rarely conducted, and collaborative settings are rarely evaluated with more than two collaborators. This impedes the

transferability of results. In some studies [138, 140], actors were employed who acted as the second collaborator in the studies. While this enhances the comparability of the results, it remains unclear how applicable the results are for real settings in which two uninstructed persons will collaborate. Furthermore, the evaluation of collaboration support features is usually limited to a user's individual (instead of the collaborative) experience. The actual purpose of collaboration support features (i.e., enhancing collaboration), is rarely evaluated. A key issue here concerns the definition and evaluation of *good collaboration* which is subject to its own research field [112, 141, 152].

Use Case Analysis. Applying XR is deemed supportive in various scenarios which have individual requirements but share certain tasks. XR's potential will be leveraged most efficiently if it can be used for different use case while allowing to perform the same tasks in similar ways. Thus, it should be analyzed which tasks need to be accomplished by which users in which use cases. The results of the use case analysis can help to establish a task taxonomy and to assess the suitability of XR technologies for a specific scenario. Furthermore, unveiling similarities across use cases can help prioritizing future research efforts.

User Onboarding. XR technologies offer modalities for spatial interaction which is different from both, natural interaction in the real world and interaction with common 2D UIs. The effective usage of XR in practical settings thus requires novel interaction paradigms but also appropriate onboarding methods. In contrast to HMDs which are designed especially for XR and are unfamiliar to most users, HHDs are already commonly used for non-XR purposes. Since existing well-established interaction paradigms might not apply in XR, users do not only have to learn the new interaction paradigm but must also unlearn old interaction paradigms. In this context, it needs to be investigated how HHDs can be used for non-XR and XR purposes and how interaction paradigms can be trained appropriately. This may also involve researching different training methods depending on the use case, technology and level of expertise.

3.2.2.2 Scalability Across Different Devices

Intuitive Use. Users should be provided with visualization and interaction techniques that are easy and intuitive to (re)learn. In this context it needs to be researched how

interaction techniques can be mapped between 2D and 3D displays. This involves investigations on whether and how intuitiveness is affected by previous experiences and well-known interaction paradigms.

Interpreting Device Data. Collaboration support features rely on tracking human behavior through sensors incorporated in XR devices. For example, the device orientation of HMD users can be used as a proxy for their head orientation and thus to generate virtual user representations for remote collaborators. However, this feature might not be as easily transferrable to HHDs whose orientation does not always correspond to the user's actual gaze directions. To prevent misunderstandings, it needs to be investigated how data obtained from XR devices has to be interpreted in different settings.

Consistent Input and Output. Mapping input and output from HMDs to HHDs and vice versa is not straightforward. When researching mapping strategies, newly gained insights concerning intuitive use, the interpretation of device data, and use case analysis should be taken into account. To enable seamless switching among different devices, available input modalities should be leveraged in a way which allows users to interact with virtual elements in a similar manner with both HHDs and HMDs. When mapping output across devices, varying display sizes of HHDs and HMDs should be taken into account. Multiple researchers have pointed out the disadvantage of small displays which are prone to occlusion when large virtual elements are rendered and impede the discoverability of virtual elements outside their reduced field of view.

3.2.2.3 Scalability Across Different Degrees of Virtuality

Scene Updates. To maximize usability in XR^S, trade-offs between visual quality and latency need to be investigated while replicating real components. In particular, it needs to be investigated which parts of a physical scene need to be reconstructed and updated in which time intervals. For example, in large XR^S environments, real-time reconstruction might not be needed for the entire scene but only for specific parts.

Reconstruction Techniques. Previously proposed techniques for virtually reconstructing physical scenes offer different benefits and drawbacks. Applying XR^S in practical settings requires techniques that allow reconstructing physical scenes which might

include different surface materials or lightning conditions. Thus, further research should focus on developing reconstruction techniques that are applicable to a variety of physical scenes.

Bidirectional Interaction. So far, research has focused rather separately on reconstructing physical scenes and their semantic segmentation. To provide options for bi-directional interaction (i.e., on-site and off-site users can reference and manipulate scene components of the XR^S space), future research should take into account the integration of results in each of these research fields to generate consistent and accessible scenes.

3.2.2.4 Scalability Across Different Numbers of Users

Access Control. Different approaches for handling access control in collaborative settings have been proposed. These include simultaneous manipulation as well as locking objects for other collaborators while one collaborator is manipulating the object. As the number of collaborators increases, simultaneous manipulation could cause unintended actions (e.g., moving objects too far). Therefore, it should be investigated for which numbers of collaborators and which use cases simultaneous manipulation is feasible and in which ownership should rather be given to one user.

Multi-Viewpoint Handling. As the number of collaborators increases, users will no longer be able to stand next to each other. Especially in co-located settings, users may thus not have the same perspective to the XR^S scene. Instead, they might stand around virtual augmentations in circles and face virtual augmentations from different viewpoints. To maintain collaboration support, it should be investigated whether and how visualizations need to be adjusted.

Awareness Cues. Various kinds of awareness cues have been proposed which indicate where collaborators are located and what they do. However, user studies in most research papers on this topic are limited to two collaborators. However, as the number of collaborators increases, simply adding the same awareness cues for all collaborators may no longer be appropriate. Instead, each collaborator should be provided with those awareness cues delivering the information relevant for this collaborator. Since tasks and cognitive load may vary among collaborators, individual configurations of awareness cues should be taken into consideration.

3.2.3 Topics Addressed in this Dissertation

In the course of this dissertation, many yet not all the research topics from the agenda are addressed. The following part of this chapter presents the design of an *XR^S Framework* which builds up on the results of a *Use Case Analysis*. With respect to scalability enhancements, this dissertation has a strong focus on the design of scalable collaboration support features and interaction techniques. Chapter 4 addresses the design of *Awareness Cues* for settings with diverse XR technologies and varying group sizes. In this context, we also investigate the *Interpretation of Device Data* in a detailed study on HHD user behavior. Chapter 5 is focused on the development of scalable interaction techniques involving research on *Intuitive Use* and *Consistent Input* methods. Taking into account the results from the *Use Case Analysis*, two practical applications are presented in Chapter 6. Initial recommendations for *User Onboarding* and teaching the novel interaction paradigms are derived from the results in the user studies conducted and summarized in the corresponding chapters.

3.3 The Basis: A Framework for XR^S

Laying the foundation for XR^S, this section presents the design of a corresponding framework. To this end, we first define a set of abstract use cases which leverage XR's key benefits and can be combined to describe specific XR applications. Next, functional and non-functional requirements are defined, based on which the framework design solution was developed. Eventually, theoretical walkthroughs demonstrate how the framework could be applied in different use cases.

3.3.1 Deriving Abstract Use Cases

To understand and specify the context in which XR^S is used, we abstract the specific use cases of XR summarized in Chapter 3.1.1 and group them into five high-level use cases that leverage XR's key benefits (see Fig. 3.3). These abstract use cases serve as blueprints for specific use cases for which the application of XR was deemed promising in previous work. The XR^S framework is designed based on the abstract use cases, such that it can be adapted to any specific use case that can be described by one or a combination of multiple blueprints.

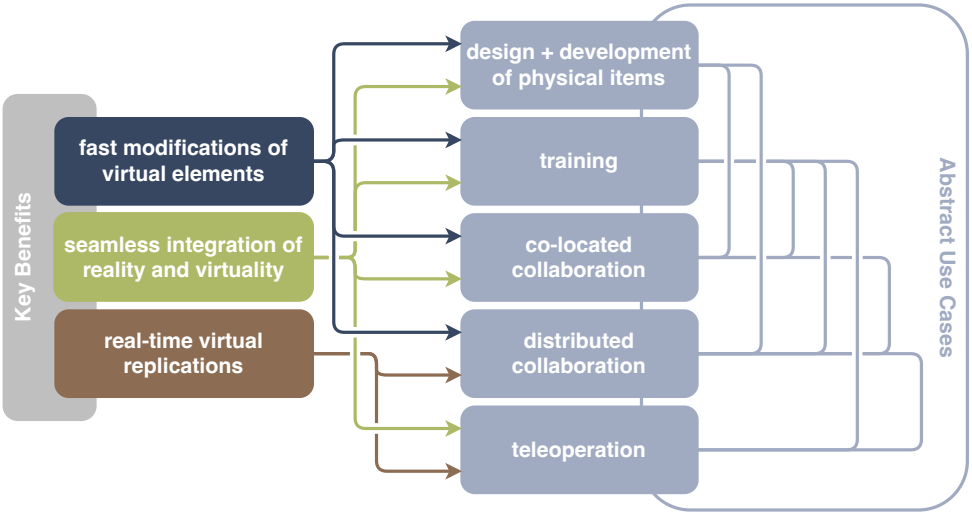


Figure 3.3: The arrows on the left side display how XR's key benefits are leveraged by the derived abstract use cases. Possible combinations of abstract use cases are depicted by the connecting lines on the right side.

The first derived use case concerns the *design and development of physical items*. Such physical items can differ in size and complexity and can be found in various domains. For example, it can refer to product design in the automotive or aerospace industry. These applications benefit from the digital nature of virtual prototypes which can be modified quickly and thus allow savings in time and material costs. In later stages of the development process physically existing parts of the prototype can be seamlessly augmented with virtual elements of the missing parts. A similar principle can be applied when planning new or restructuring existing facilities such as manufacturing sites but also private residencies.

The second abstract use case concerns *training* scenarios which can range from surgery to machine operation. Purely virtual training environments can be set up multiple times allowing trainees to practice in parallel. In MR, the seamless integration of virtual elements into the real environment prevents trainees from having to shift their focus between multiple sources and allow them to concentrate on the actual training task. Furthermore, XR can improve the accessibility of training environments by integrating virtual simulation and thereby reducing safety issues, enabling training without permanent supervision. Thereby, fast modifications of the virtual augmentations allow adapting the level of information and reducing the degree of virtuality in correspondence with learning progress.

Furthermore, *co-located collaboration* and *distributed collaboration* are considered as abstract use cases. Both benefit from the fast modifications of virtual elements enabling customized accesses to the XR space which fits their responsibilities, experience, and

personal preferences. Communication among the collaborators is supported by virtual elements that display user activities. In co-located scenarios, these can be seamlessly integrated in each collaborator's field of view whereas remote collaborators may be provided with a virtual replica of the on-site environment.

Similarly, *teleoperation* can be enabled through XR interfaces, the fifth abstract use case. Depending on the distance between the operator and the machine, a VR application (i.e., a virtual replica of the real machine and its environment) or a MR application (i.e., virtual augmentations that are seamlessly integrated in the physical machine's environment) may be used. In both cases the XR UI allows reviewing effects of a command in virtual simulations prior to execution.

The identified abstract use cases can also be combined with each other. For example, product development can be performed by a single user, in collaboration with co-located and/or distributed collaborators. Teleoperating machines can be subject to a training scenario, or robotic arms can be teleoperated to allow remote collaborators in distributed scenarios to manipulate physical objects on site. Collaboration can also be applied in training scenarios such that the trainee can be supported by remote or co-located collaborators who supervise the trainee and may intervene if necessary.

3.3.2 Defining Requirements

Based on the abstract use cases identified in Chapter 3.3.1, we formulate the following functional and non-functional requirements for the XR^S framework. Thereby, we use the terminology introduced in Chapter 2.1. Thus, we use the term *virtual component* to refer to shared scene components which are exclusively virtual in both the on-site environment and the off-site environment. The term *real component* is used to refer to shared scene components which appear as physical objects to on-site users and as virtual replicas of these physical objects to off-site users.

3.3.2.1 Functional Requirements

First, functional requirements for the XR^S framework are specified that concern the hardware and technology used to access the XR^S space, the available interaction modalities, and the visualization of user locations and activities as well as the visualization of the real and virtual components.

Access

REQ 1 On-site users can access the XR^S space via a MR-HMD or a MR-HHD.

REQ 2 Off-site users can access the XR^S space via a VR-HMD.

Interaction

REQ 3 On-site users can reference real components.

REQ 4 On-site users can reference virtual components.

REQ 5 On-site users can manipulate real components.

REQ 6 On-site users can manipulate virtual components.

REQ 7 Off-site users can reference real components.

REQ 8 Off-site users can reference virtual components.

REQ 9 Off-site users can manipulate real components.

REQ 10 Off-site users can manipulate virtual components.

Visualization

REQ 11 Each collaborator sees where the other collaborators are.

REQ 12 Each collaborator sees what the other collaborators do.

REQ 13 Off-site users are provided with a virtual replica of static real components.

REQ 14 Off-site users are provided with a virtual replica of dynamic real components.

REQ 15 On-site users are provided with visual representations of virtual components that are seamlessly integrated into the physical scene.

REQ 16 Off-site users are provided with visual representations of virtual components that are seamlessly integrated into the virtual scene.

3.3.2.2 Non-functional Requirements

To apply XR^S spaces in the real world, usability must be maintained across different system configurations. Therefore, the following non-functional requirements are defined.

REQ 17 Users can intuitively switch between devices.

REQ 18 Users can intuitively switch between degrees of virtuality.

REQ 19 Usability is maintained with an increasing number of collaborators.

REQ 20 The interaction techniques for manipulating and referencing real and virtual components provide high usability.

3.3.3 Designing a Solution

Based on the defined functional and non-functional requirements, a conceptual framework for XR^S spaces was developed (see Fig. 3.4) that incorporates the following system features.

3.3.3.1 Access Points and Data – REQs 1, 2, 17, 18

The designed framework provides three access points to an XR^S space: MR-HHDs, MR-HMDs, and VR-HMDs. To this end, first a virtual replica of the static scene components is generated which builds the basis for the VR environment in which off-site users access the XR^S space. Then, virtual replicas of dynamic real components as well as of on-site and off-site users are generated and integrated into the VR scene. Similarly, the MR scene is augmented with virtual replicas of off-site users. Purely virtual components are integrated in both the MR and VR scene. Throughout a session, each collaborator's application reads and writes data from and to a database storing information about each user, real component, and virtual component.

3.3.3.2 Subscribing to Collaborators – REQs 11, 12, 19

To enable effective task completion and collaboration, it is important to provide all users with relevant information while avoiding information overload. Therefore, the users can individually subscribe to receive information from other collaborators.

The database stores each user's id, role (on-site or off-site user), activity (referencing or manipulating an object), position and orientation, as well as their subscriptions to the location and activities of other collaborators.

Depending on a collaborator's subscription, the other collaborators' locations are visualized through avatars and their activities are represented by awareness cues. These subscriptions can be configured before the start of the collaborative session and adjusted individually during the session.

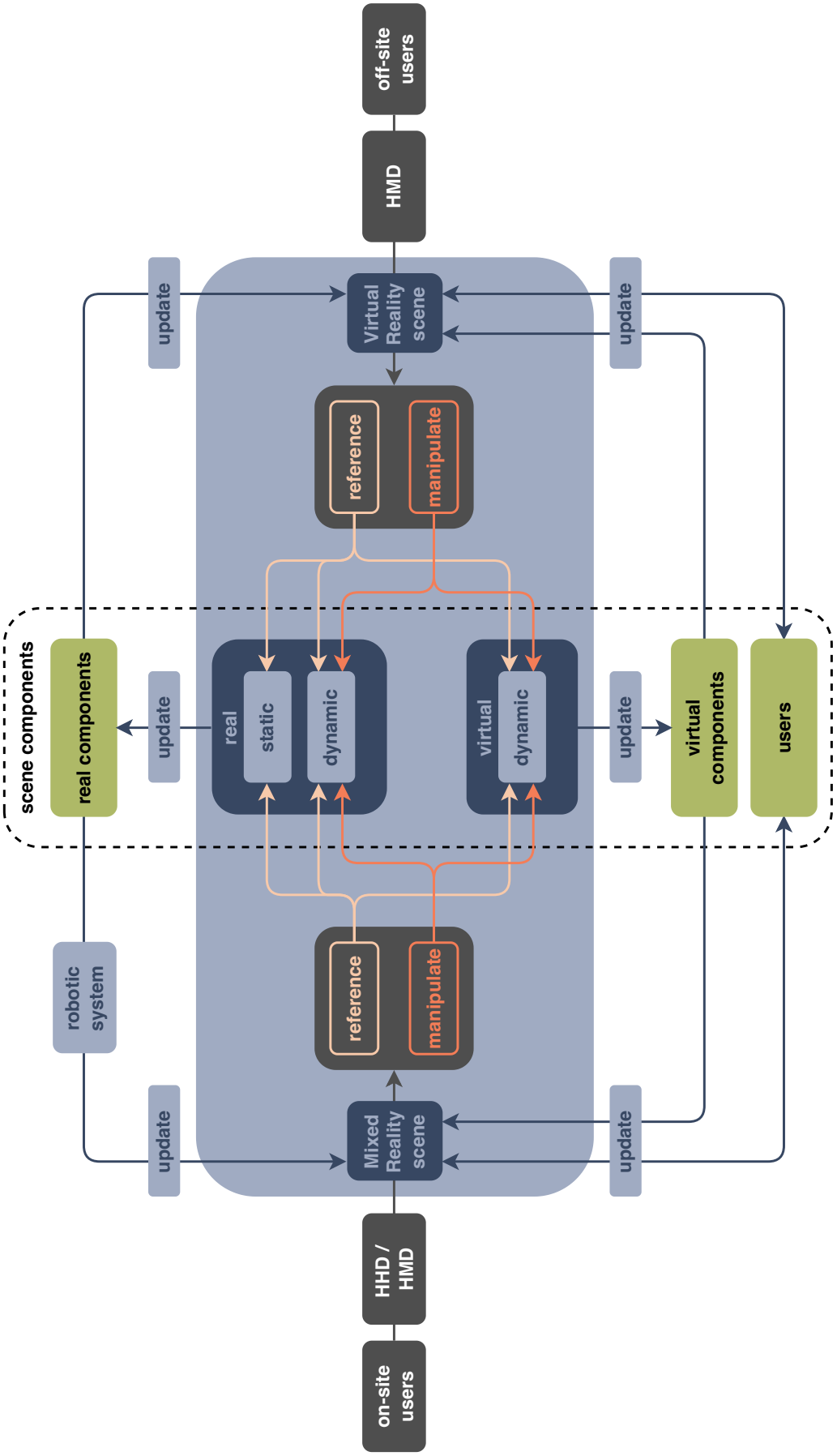


Figure 3.4: XR^{S} framework.

3.3.3.3 Visualizing Static Scene Components – REQ 13

Static scene components are not meant to change their position or orientation during the session (e.g., the room in which the on-site collaborators are located). To enable full scalability regarding different degrees of virtuality, off-site users must be provided with a virtual replica of these static components. To this end, appropriate reconstruction techniques must be selected in accordance with the required quality-latency trade-off. Static components which require no updates during the session should be virtually replicated with the highest quality. In this context, semantic segmentation of scene components also becomes relevant if users should be able to reference single parts of it.

3.3.3.4 Visualizing Dynamic Scene Components – REQs 14, 15, 16

Dynamic scene components are manipulable and thus can require frequent updates during the session. They include both purely virtual components and real components which appear as physical objects to on-site users and as virtual replicas to off-site users. Users are not classified as dynamic components and are treated separately as they possess more properties that may change during the session than objects.

Physically existing components are only visualized for off-site users. In contrast to static scene components, they pose different requirements to the quality-latency trade-off. Their virtual counterpart must be updated more often such that low latency may be prioritized over high quality. The choice of the reconstruction technique may also be influenced by the object's properties such as the surface material. If only the object's position and orientation will change during the session, it can be replicated in advance and integrated into the off-site user's environment. In this way, tracking during runtime is limited to the object's position and orientation. However, if the object's appearance will change during the session, this must be tracked and updated accordingly during runtime.

Visualizing dynamic virtual components (i.e., components which are purely virtual in both the MR and VR scene) is less complex as their properties can be stored and updated in the database based on user interactions. Then, clients can read the same information about an object (e.g., position, orientation, size, status, or current owner) from the database.

3.3.3.5 Visualizing User Location and Activity – REQs 11, 12

To support effective co-located and distributed collaboration, every user needs information regarding their collaborators' locations and activities. While in MR, collaborators can see each other naturally, users that access the XR^S space in VR need to be provided with real-time virtual replicas of their collaborators. Similarly, on-site users in MR need virtual replicas of their remote collaborators who are located off site.

Apart from information on a user's location, both on-site and off-site users need to be provided with cues which display the other collaborators' current activities (e.g., referencing or manipulating objects). The visual representation of the collaborators and their activities is adapted in correspondence with each user's individual subscriptions stored in the database.

3.3.3.6 Referencing Scene Components – REQs 3, 4, 7, 8

Effective communication requires the option for users to reference scene components (e.g., pointing at them as common in face-to-face communication). In XR, a specific action can be executed which highlights the referenced scene component for other collaborators depending on their subscription. This can be implemented in different ways. For example, by adapting the object's visual representation (e.g., by changing the color of a virtual component or of a real component's virtual replica in VR or by adding virtual overlays to a real component in MR) or by playing 3D audio. Users can also reference objects to select and then manipulate them.

3.3.3.7 Manipulating Dynamic Scene Components – REQs 5, 6, 9, 10

Our framework integrates manipulation of real and virtual components in terms of translation and rotation for all access points of XR^S. This scalability is crucial as users should be able to switch between the access points without losing options for operation.

The manipulation of virtual components requires appropriate input techniques such that the virtual component's updated position and orientation can be computed, shared, and its visual representation can be adapted accordingly.

Manipulating real components in the on-site environment is more complex. In this con-

text, we propose the integration of a robotic system that allows users to remotely manipulate real components on site. Off-site users can manipulate the virtual replica of a real component. Then, they can command the robotic system to perform the same task in the on-site environment. To this end, the new position and orientation of the object is shared with the robotic system via the database.

Similarly, on-site users can benefit from the robotic system when manipulating large, heavy, or hazardous real components. In this case, virtual overlays of the real components could be manipulated to instruct the robot. In the case of a training session, the connection with the robotic system could be disabled or enabled depending on the trainee's learning progress.

3.3.3.8 Scalable Interaction Techniques – REQs 17, 18, 20

To enable seamless switching between the three access points to XR^S, the interaction techniques for referencing and manipulating real and virtual components must stay consistent between MR-HHDs, MR-HMDs, and VR-HMDs. Specifically, we refer to scalable interaction techniques as those that prevent large overheads of cognitive and temporal efforts to adapt to a new access point (i.e., switching between access points should be as intuitive as possible). The full scalability regarding options of interaction between on-site users and off-site users as implemented in our framework builds the basis for the integration of such scalable interaction techniques.

3.3.4 Walkthrough

In the following, two potential applications of our framework are demonstrated through theoretical walkthroughs. Both example use cases (i.e., collaborative prototyping and training and teleoperation) can be described by a combination of the high-level use cases introduced in Chapter 3.3.1.

3.3.4.1 Example 1: Collaborative Prototyping

Designing and developing physical items, such as cars or aircrafts, typically involves co-located or distributed collaborators from different fields. Throughout the process, these collaborators have different tasks and responsibilities which may require different levels of

information. In this context, the presented framework can help providing the collaborators with the required information exactly when and where it is needed.

During the ideation phase, i.e., the beginning of the development process, users may be immersed in a completely virtual environment and develop a first prototype. As long as no physical parts of the product (i.e., real components) exist, there is no on-site environment and all users join the XR^S space in a VR scene. Within this scene, dynamic virtual components (i.e., the parts of the product being developed) can be referenced and manipulated. Thereby, each collaborator may subscribe to other collaborators to receive information on the other collaborators' locations and activities. At the same time, data on each user's position, orientation, and activity is written to the corresponding database and retrieved by the collaborators in accordance with their subscriptions.

As soon as the first physical parts of the product are available, collaborators who are co-located on site may switch to a MR scene which integrates the physically existing real components with virtual components representing the missing parts. This setting allows reviewing different virtual configurations of the missing parts in iterative prototyping stages. Thereby, decisions may be made based on key parameters which could be generated in real time by digital twins. On-site users can thereby reference and manipulate the real and virtual components as described by the framework.

Collaborators who are located off site can join the XR^S space through the same VR scene as before. The basis of this VR scene is generated by a virtual replication of the static real components. Furthermore, the properties of dynamic real components are tracked such that their virtual replica can be integrated into the VR scene. These components are updated together with the exclusively virtual components which are also continuously tracked. Then, off-site users can reference and manipulate both virtual replicas of real components as well as exclusively virtual components like described in the framework.

Similar to the exclusively virtual space at the beginning of the development process, both on-site and off-site users may subscribe to receive information about each other's location and activities.

Throughout the iterative stages of the development process the degree of virtuality decreases until the physical prototype is only augmented with single virtual components. Apart from this continuously changing degree of virtuality the degree of virtuality may also change within a development stage if on-site users become off-site users or vice versa depending on their location. Furthermore, in practice, one person can be involved in

several design and development processes of different products at the same time. As the current development status of these products may vary, this person may have to switch between devices and degrees of virtuality several times a day. Our framework implements multidimensional scalability enhancements to reduce the temporal and cognitive efforts when users have to switch between these technologies while allowing them to keep focusing on the actual task.

3.3.4.2 Example 2: Training and Teleoperation

To ensure the correct and safe execution of complex and hazardous tasks, employees must complete appropriate training in advance. However, in practical settings, the relevant machines may be in operation or occupied by other trainees, limiting training possibilities. In addition, some tasks may require the supervision of experts whose availability is limited as well.

The need for supervisors can be reduced by transferring training to virtual scenarios in which safety-critical parts are eliminated. Furthermore, the replication of these virtual training environments allows parallelizing the training of multiple trainees, provided there is sufficient hardware available.

The application of the XR^S framework can allow trainees to start in a VR scene which provides them with a virtual replica of the actual working environment such that operations can be practiced by manipulating virtual replicas. In this case, interaction with the virtual replicas of real components should not be executed in the real working environment. As learning progresses, trainees may switch to MR scenarios in which they can practice operations in the real working environment. At the same time, safety-critical parts can still be virtualized. Thereby, scalable interaction techniques that rely on the same interaction paradigms as in the VR scene, allow them to focus on the actual task (i.e., the operation to be learned). When training is accomplished and they move on to the operation in the real world, single virtual elements may still be integrated in the real scene for initial assistance.

At the same time, a novice can request help from a remote expert. The remote expert who is located off-site may join the XR^S space in a VR scene which is based on a virtual replica of static real components that is augmented with virtual replicas of dynamic real components and exclusively virtual components in real time. Depending on the specific task, the remote expert's interactions with the virtual replicas may be executed on site

through the robotic system. In this case, the remote experts can act as teleoperators. The framework provides similar UIs for remote assistance and teleoperation tasks which are likely to be performed by the same person.

Parts of this chapter have been previously published in:

V. M. Memmesheimer and A. Ebert (2022): Scalable Extended Reality: A Future Research Agenda. Big Data and Cognitive Computing, 6(1):12. doi: 10.3390/bdcc6010012.

V. M. Memmesheimer and A. Ebert (2023): A Human-Centered Framework for Scalable Extended Reality Spaces. In J. C. Aurich, C. Garth, and B. S. Linke (Eds.) Proceedings of the 3rd Conference on Physical Modeling for Virtual Manufacturing Systems and Processes (IRTG 2023), pp. 111–128. Springer, Cham. doi: 10.1007/978-3-031-35779-4_7.

Chapter 4

Scalable Collaboration Support Features for Extended Reality

In the previous chapter we present a concept and a framework for XR spaces that integrate scalability enhancements regarding different degrees of virtuality, different devices, and different numbers of potentially distributed users. The realization of such XR^S environments requires further research in several areas. These include scalable collaboration support features which will be in the focus of this chapter. More specifically, this chapter deals with the design of visual cues which support collaboration in settings which involve different XR technologies and varying group sizes.

After a summary of previous research and aspects related to the development of collaboration support features for XR settings in Chapter 4.1, we discuss the scalability of existing collaboration support features in Chapter 4.2. To this end, we consider the nine different collaboration styles introduced in Chapter 2.2 and define scalability objectives based on related requirements of the XR^S framework. Identified scalability limitations are translated into two research questions. The first research question deals with the matter of transferring data obtained from an HHD into a correct representation of its user's activities whereas the second research question focuses on the avoidance of visual overload as the number of collaborators increases.

Addressing the first research question, Chapter 4.3 presents the results of a detailed study on how different user poses, display sizes, and device orientations influence an HHD user's behavior and the obtained device data. The second research question is addressed in Chapter 4.4, which presents the design of mechanisms for individually activating the visibility of awareness cues. Chapter 4.5 provides a joint discussion of the results.

4.1 Related Research and Aspects

While XR technologies are deemed supportive for collaboration they also pose challenges regarding the conveyance of user behavior and activities among the collaborators. In this context, a lot of research has been conducted on how communication and cooperation among collaborators can be supported and made as similar as possible to face-to-face collaboration. More specifically, avatars were augmented with further awareness cues to make user activities inside XR environments and interactions with its components visible for their collaborators.

Related research has considered and compared the combination of various cues for visualizing a user's viewing direction. For example, Piumsomboon et al. [139] compared three awareness cues (i.e., a field of view frustum, a field of view frustum combined with a ray originating from the head, and a field of view frustum with a ray originating from the eye) against a baseline (i.e., virtual head and hands without further cues) and found that the combination of view frustum and head-ray was most useful.

Further research on gaze visualizations was presented by Jing et al. [84] who compared unidirectional gaze (i.e., each user sees only the gaze visualizations of the other collaborator) and bi-directional gaze (i.e., each user sees their own and the other collaborator's gaze visualization) combined with different visualizations displaying the current gaze behavior state (i.e., browsing state, focus state, and joint state). They report that bi-directional gaze including gaze behavior was the preferred condition. Similar findings were reported by Bovo et al. [17] who compared different variants of visual cues which display a user's viewing direction on 2D surfaces based on average fixation maps. Their results show that bi-directional cues, that is being able to see both the own and the collaborator's view cue, increases confidence in the cues and can reduce the need for checking the collaborator's avatar to understand the actions.

Kim et al. [91] compared the combination of different cues for conveying hand movements, i.e., a remote user could provide instructions to a local user through a virtual hand only, a virtual hand with pointer-ray, a virtual hand which can make virtual sketches in mid-air, and a virtual hand with both a pointer-ray and the sketching option. The pointer and sketch were automatically enabled if a specific hand gesture was detected. The remote user then guided the local user in an assembly task whereby the remote user's scene view depended on the local user's view. They report that participants preferred the sketch cue which also improved the overall performance. Furthermore, they report that additional

cues (i.e., pointer and sketch) increased mental effort. In a follow-up study [92] they compared the cues in the dependent view setting to an independent view setting (i.e., the remote user accesses the scene independently). It turned out that for the local user it was much more difficult to correctly interpret the hand gesture of the remote user with the independent view than with the dependent view setting. Similarly, the sketching option was only considered useful with the dependent view. On the contrary, the pointer-ray was considered more useful with the independent view setting than with the dependent view. Furthermore, XR-supported collaboration systems which allow conveying information about hand movements through laser pointers were deemed positive in previous research [138, 144].

Bai et al. [5] compared awareness cues for conveying both hand gestures and viewing direction of a remote user to a local worker. Three conditions of awareness cues were compared: virtual hands which mimic the hands of the remote user, a gaze-ray displaying a remote collaborator's gaze direction, and the combination of hand and gaze-rays. In the study, the cues augmented a simple avatar head with a view frustum. Additionally, a virtual arrow was added to the scene which is pointing to the avatar to help users finding each other. It was found that the majority of participants preferred the combination of hand gestures and gaze rays over either of them alone. Similarly, Bovo et al. [17] note that view cues and hand pointers complement each other. In fact, the view cue can be used as a confirmation that the other collaborator is seeing the hand gesture (i.e., if it is in the field of view). These findings are in line with the recommendation in the survey on communication cues for remote guidance from Huang et al. [76]. They distinguish between explicit and implicit communication cues. Explicit communication cues are grouped into pointers, virtual annotations, hand gestures, or virtual models of known task objects. Implicit communication cues refer to facial expressions, eye-tracking, or body posture. They conclude that adding these implicit cues to the explicit cues effectively supports communication among collaborators.

Previous research has a strong focus on the development of collaboration support features for HMDs. While some research papers consider settings that involve both HHDs and HMDs [57, 96, 133, 56, 107, 175, 44], they are rather focused on providing technical solutions for collaborative cross-device settings than on evaluating the scalability of awareness cues. In the context of collaborative learning applications the guidelines from Drey et al. [44] emphasize the importance of providing equivalent user representations and options for interaction for all collaborators even when they use different devices.

Regarding visual representations of HHD user behavior, it has been proposed to display rays that emerge from the HHD upon touch input (e.g., [57, 176, 107]). However, touch input is considered unfavorable for MR-HHDs [62] since this requires holding the device with one hand and is thus prone to fatigue and scene occlusion. This is avoided if the ray emerges from the device center such as in [96]. The system from Vanukuru et al. [176] for remote collaboration of two HHD users integrated more advanced user representations. Thereby, front facing depth cameras capturing the users were attached to the HHDs. In this way, 3D holograms or spatial videos of remote collaborators could be integrated in the shared space. Interestingly, it was reported that some of their participants found the 3D user holograms uncanny.

Outside the context of collaborative XR systems, research has further explored tracking users through an HHD's front camera [2, 4, 179, 111, 78, 124]. For example, Hueber et al. [78] track an HHD user's head rotation through the front camera to extend touch input. Depending on the direction in which the head is rotated, specific actions can be executed. Voelker et al. [179] use the cameras of HHDs for estimating the gaze direction of collaborators in a non-XR setting. Ahuja et al. [2] used inverse kinematics to generate an avatar of an HHD. Thereby, they also capture the user's head through the front camera to adapt the avatar accordingly. Furthermore, combing head tracking through the front camera with world tracking through the rear camera was proposed to estimate at which part of the real world the user is looking [111, 124] as well as for enhanced tracking of the user's hand, head, and body motion [4]

4.2 Objectives

Previous research mainly evaluated awareness cues in settings with only two collaborators and has a strong focus on awareness cues for HMD users (e.g., [5, 84, 92, 139, 138, 140]). XR^S as introduced by us, however, seeks to enable collaboration in cross-device settings which scale with increasing group sizes and involve VR and MR as well as HHDs and HMDs.

Effective methods for capturing and visualizing each collaborator's activities and location in XR are required not only for distributed but also for co-located collaboration. While co-located collaborators are usually able to see where the other collaborators are located, it can be difficult for them to follow the other collaborators' activities, for example when

they reference or manipulate virtual components in the MR scene. Thus, the collaboration support features designed in this chapter contribute to the following requirements of XR^S from Chapter 3.3.2.

- **REQ 17** Users can intuitively switch between devices.
- **REQ 18** Users can intuitively switch between degrees of virtuality.
- **REQ 19** Usability is maintained with an increasing number of collaborators.

As an initial step towards appropriate collaboration support features for XR^S these requirements are specified in more detail, yielding the following set of objectives.

- **Objective 1** Each collaborator can see the location and orientation of all other collaborators.
- **Objective 2** Each collaborator can see the activities of all other collaborators.
- **Objective 3** The awareness cues represent each collaborator's behavior correctly.
- **Objective 4** All collaborators are equipped with the same set of awareness cues.
- **Objective 5** All collaborators can intuitively control their own awareness cues.
- **Objective 6** All collaborators can match all visible awareness cues to the corresponding collaborator.
- **Objective 7** All visible awareness cues support the collaborative work process.

In the following we discuss how common awareness cues align with these objectives under the consideration of varying group sizes and the nine different collaboration styles introduced in Chapter 2.2 (see Fig. 2.1). This involves synchronous co-located and distributed collaboration using three types of XR technologies (i.e., MR-HHDs, MR-HMDs, and VR-HMDs). Regarding collaboration support features, we consider avatars which are augmented with two popular awareness cues: head-rays (i.e., rays originating from a user's or avatar's head) and hand-rays (i.e., rays originating from a user's or avatar's

hand). As a default scenario for evaluating these cues, we assume that both head-rays and hand-rays of all collaborators are permanently enabled.

Even when the group size of a collaborative setting increases, collaborators are – theoretically – still able to see the location and orientation (**Objective 1**) as well as the activities (**Objective 2**) of the other collaborators based on the provided avatars and awareness cues. While the correct representation of user behavior through these awareness cues (**Objective 3**) is not affected by an increasing group size, it is heavily affected in cross-device settings. In contrast to HMD users whose orientation in space can usually be derived from the HMD's orientation, an HHD's orientation does not necessarily display the actual orientation of the user at all times. Similarly, the awareness cues displaying the user's activities may be misleading if the HHD user is not interacting with the HHD while awareness cues are still being generated. Hence, in cross-device settings collaborators can see the awareness cues which display the other collaborators' location and orientation (**Objective 1**) as well as their activities (**Objective 2**) but they might not represent user behavior correctly (**Objective 3**) and could thus fail to support the collaborative work process (**Objective 7**).

Implementing head-rays and hand-rays for VR-HMDs, MR-HMDs, and MR-HHDs requires knowledge about the position and orientation of each collaborator's head and hand. The generation of head-rays for HHD users is more challenging than for HMD users and was not taken into account in [57, 96, 133]. As described in Chapter 3.1.2, inverse kinematics have been used to adapt the pose of avatars for HHD users based on the HHD's orientation [122, 105, 2]. Outside of the context of XR-supported collaboration, it has been considered to use the device's front camera to capture an HHD user's face [2, 4, 179, 111, 78, 124]. However, this approach presupposes that the front camera is accessible which might not always be the case.

Further challenges arise in the generation of hand-rays for HHD users. In [57] the HHD user is represented by a basic avatar and a virtual tablet. When the user touches the screen, a virtual hand appears on the virtual tablet on the position of the touchscreen that corresponds to the touched point. They also consider the temporary integration of a virtual ray originating from the tablet. Thus, the cues are only visualized upon user input. While this prevents incorrect visualizations of user activities the approach may lack implicit information about user activities. Moreover, holding the HHD with one hand as in this scenario has been deemed unfavorable in the context of spatial interaction with MR-HHDs [62] due to fatigue and occlusion issues. Here, laser pointers emerging

from the device center, such as in [96], would allow holding the device with both hands. For HMD users, the generation of hand-rays based on hand or controller tracking is more straightforward [91, 92, 193, 144]. Here a major issue, restricted tracking areas, appears to have been solved by novel HMDs such as the Apple Vision Pro which utilizes downward-facing cameras to extend the tracked space.

Thus, depending on the involved XR technologies, providing all collaborators with the same set of awareness cues (**Objective 4**) can be related to minor or major challenges in cross-device settings.

The fact that user behavior is represented correctly (**Objective 3**) does not necessarily imply that it is possible for users to intuitively control their own awareness cues (**Objective 5**). It is crucial for users to know based on which data the awareness cues are generated and how this data is processed. The cognitive effort required to control the awareness cues should be kept as low as possible. If a lot of cognitive resources are required to use the device in a way that generates data for correct behavioral representations, there may be insufficient cognitive resources for performing the actual task. This could impede the collaborative work process rather than support it (**Objective 7**).

To enable seamless switching between MR-HHDs, MR-HMDs, and VR-HMDs while allowing users to intuitively control their own awareness cues (**Objective 5**) the same user behavior should result in the same awareness cues using all devices. In this way, users do not have to adapt their behavior to control their own awareness cues when switching to another technology. For example, when hand-rays are generated by tracking hands [91, 92] or controllers [193, 144] for HMD users and based on the movement of the handheld device [96] for HHD users, the same behavior (i.e., hand movement) is consistently transferred to the hand-ray visualization as long as controllers or the HHD are held in the users hand. Consistent mapping of user behavior to head-ray visualizations requires reliable head tracking of both HMD and HHD users. While HMDs can serve as a proxy for their user's head orientation, this is not necessarily the case for HHDs and requires further investigation. For example, estimating an HHD user's viewing direction through the front camera is only possible if the front camera is accessible which may require users to adapt their behavior to ensure accurate cue generation.

Providing all collaborators with the same set of awareness cues (**Objective 4**) also poses challenges in collaborative settings with large group sizes. Enabling all awareness cues for all collaborators is likely to produce visual clutter when the number of collaborators increases. Thus, always-on awareness cues are prone to visual overload as it becomes

more difficult for the users to match the visible awareness cues to the correct collaborator (**Objective 6**). This approach may thus rather impede the collaborative work process than support it (**Objective 7**). This issue is also emphasized by the findings of Pereira et al. [133] who considered larger group sizes during their evaluation and note that the users had difficulties in identifying the source of highlight mechanisms. Hand-rays and head-rays are particularly prone to this kind of visual clutter as they take up a large part of the screen. Further occlusion issues may be evoked by the integration of avatars. In this context, Piumsomboon et al. [140] point out that their miniature avatar has the advantage of taking up less space on the screen. In addition, avatars can become superfluous in co-located environments when users can see each other. However, if several co-located collaborators are pointing at or looking at a virtual object, it may still be useful to display the corresponding ray.

From this discussion the following two main research questions emerge which must be addressed in the design of collaboration support features that scale between different XR technologies and varying group sizes.

- **Research Question 1:** How can data obtained from the HHD be transferred into correct representations of the HHD user's activities?
- **Research Question 2:** How can visual overload be avoided when the number of collaborators in a group increases?

In the following two sections, these research questions are addressed. First, a detailed study is performed to investigate the behavior of HHD users in Chapter 4.3. Then, mechanisms for automated adaptations of visual cues are designed to prevent visual overload in large groups in Chapter 4.4.

4.3 Investigating the Behavior of Handheld Display Users

While HHDs offer a cost-effective alternative to HMDs for accessing MR, the creation of accurate user representations is challenging. Compared to HMDs, an HHD's orientation in space is more likely to deviate from its user's actual viewing direction as a user may not be holding device right in front of his or her face at all times.

While equipping HHD users with additional hardware like eye-trackers worn on the head could solve this issue and help generating more accurate head-rays, this approach invalidates the main benefits of HHDs, i.e., its ubiquity and easy setup.

Addressing **Research Question 1** and as an initial step towards enhanced representations of MR-HHD users, a detailed study was conducted to explore how accurately a ray (in the following referred to as HHD-ray) originating from the center of the device determines the position of a virtual object while the user is looking at it. Moreover, the accessibility of the front camera is evaluated through face detection.

4.3.1 Setup and Experimental Design

We explored how users interact with different MR-HHD devices in different display orientations and body poses in an experiment with 20 participants (10 male/female, 11 wearing glasses). To this end, we developed a MR-HHD application in Unity using AR Foundation with ARKit and deployed it either to an Apple iPad Pro (11 inch, Gen. 3) or an Apple iPhone 14 Pro. The application displayed virtual cubes at different positions and collected data on device orientation and face detection in the background. During the experiment, we asked the participants to find and then focus on these virtual cubes until they disappeared.

We followed a within-subjects design in which we considered the MR-HHD configuration device {tablet, phone} $\times$ orientation {landscape, portrait} $\times$ pose {standing, sitting} and cube position (11 positions differing in direction and distance; see Fig. 4.1) as independent variables. The experiment took into account a set of dependent variables. We were specifically interested in how the HHD-ray's accuracy as well as the accessibility of the front camera were affected by the independent variables. Furthermore, we asked the participants to rate the discomfort of each device $\times$ orientation configuration from 1 (very comfortable) to 5 (very uncomfortable) and took notes on the manner in which the participants held the device.

At the beginning, the participants watched a video which introduced them to the overall procedure followed in the experiment. In the video, the participants were also asked to not walk around during the experiment and an explanation of the face detection feature was given. As the experiment was supposed to investigate the participants' unbiased, natural behavior while interacting with different MR-HHD configurations, the participants were

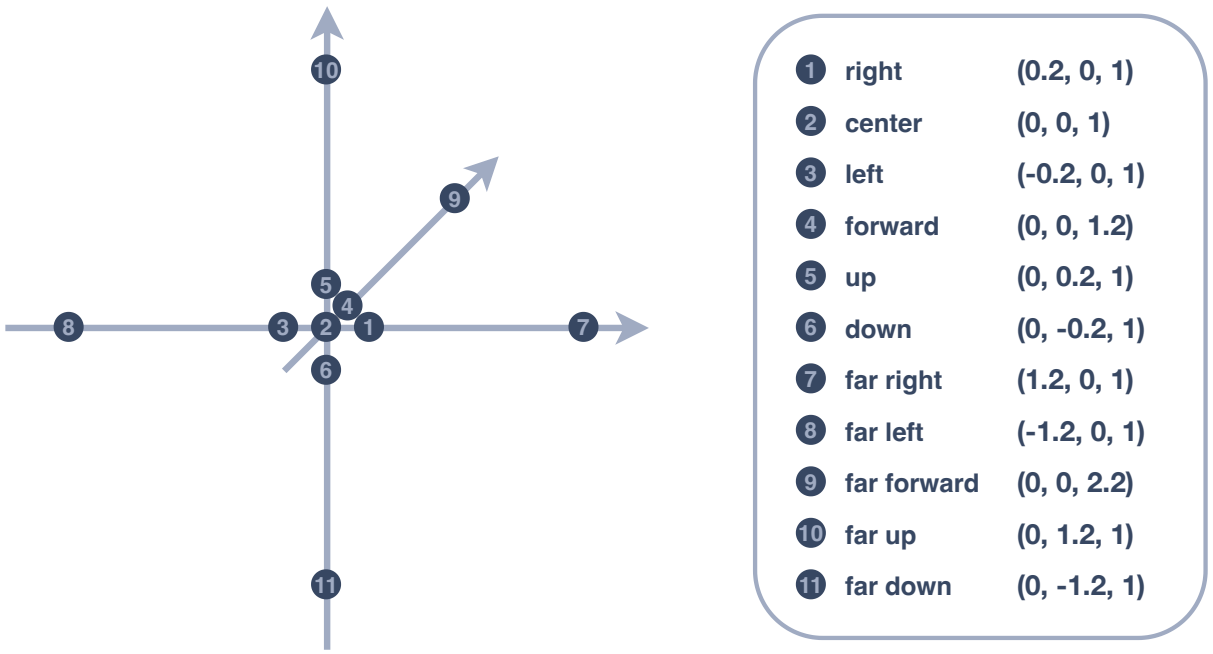


Figure 4.1: Cube positions based on the HHD's starting position on a table (approx. 1.2m high, all values in meter)

not informed about the HHD-ray.

After watching the video, each participant completed the task set consisting of 11 cubes in the 8 MR-HHD configurations (i.e., device × orientation × pose). In total, each participant thus focused on 88 cubes. The order in which a participant used the different configurations was as follows. Half of the participants performed all device × orientation configurations while sitting before standing and vice versa. Within one pose configuration, half of the participants performed the task sets with the phone before the tablet and vice versa. And within one device × pose configuration tasks were performed in landscape before portrait mode. Whenever the tablet or phone was used for the first time, a short training session was performed to ensure that the participant understood the procedure.

Within one task set, the procedure was as follows. At the beginning a virtual capsule appeared at the center of the scene. The capsule disappeared 2 seconds after it was captured by the HHD's camera. While the capsule was visible, participants were asked to focus on the capsule. After the capsule disappeared, they had to search for the first virtual cube (0.15m × 0.15m × 0.15m) and focus on the cube. 5 seconds after the cube was captured by the device camera, it disappeared and the capsule appeared again in the scene's center. As before, the capsule appeared for 2 seconds after being captured by the HHD's camera. In this way, the participants had to return to a neutral position

before the next cube appeared. This procedure was followed for all 11 cubes. Throughout the experiment, the HHD's front camera was always located at the top of the display in portrait mode and at the left side in landscape mode.

To investigate how the accuracy of the HHD-ray is affected by the direction in which the cube is located relative to the participant, the cubes were placed on the left/right side of the user, above/below the user, and in front of the user. Furthermore, we wanted to investigate whether the distance from the user to the cube affects the HHD-ray's accuracy. We therefore considered two cube positions in each of the directions (e.g., left and far left). Cubes were not placed behind the user as these positions would be equivalent to the positions in front of the user when he or she turns around. Based on the HHD's starting position on a table (approx. 1.2m high), the cubes were set to the 11 positions marked in Fig. 4.1.

To evaluate the accuracy of the HHD-ray, we cast an invisible ray from the main camera (i.e., the display's center). In the background, the MR-HHD application then constantly checked if the ray hit a semi-transparent plane which surrounded the currently visible cube. The position of this plane (size: 1.5m $\times$ 1.5m) was always set to the same position as the currently visible cube and the plane's orientation was constantly adapted to match the one of the HHD. In this way, the HHD-ray's accuracy was assessed by the distance between the center of the cube and the hit point of the ray on the plane. As such, the HHD-ray was more accurate the closer the cube was to the center of the display.

As described above, an HHD's front camera could also be used to enhance avatars and collaboration support features. However, this presupposes that the HHD's front camera is accessible and users do not cover the front camera with their hands. Thus, the integration of such features may affect the way a user holds the HHD. To this end, we integrated a basic version of such a feature and constantly checked if ARFaces from Unity's AR Foundation could be detected. Red borders appeared on the screen whenever face detection failed. Participants were informed about this feature before the experiment and asked to adjust the way they held the device as soon as the red borders appeared.

4.3.2 Results

The data obtained for the HHD-ray's hit points on the semi-transparent plane surrounding the currently visible cube show that the HHD-ray hit the plane in more than 99% of the

frames. Thus, the frames in which the HHD-ray did not hit the plane were omitted in Figs. 4.2, 4.3, and 4.4.

Both the mean distances between the HHD-ray's collision points on the semi-transparent plane and the cube (see Figs. 4.2, 4.3, and 4.4) as well as the heatmap displaying the 2D hit points around the center cube (see Fig. 4.5) show that the HHD-ray was particularly accurate for the reference cube placed at the center position.

In general, the participants tended to center the cubes on the screen during our experiments. However, there was also a tendency to stop moving the device towards the cube earlier as the distance from the neutral position (i.e., the user's and device's position while capturing the capsule) to the cube increased. In these cases, centering the cube in the screen would require more physical effort.

For instance, if the cube appeared at the far left position, the participants tended to move the device to the left and stopped when the cube appeared at the left side of the display. Since the cube is then not located in the center of the display, the HHD-ray's accuracy decreases. This user behavior was observed for movements to the left and right as well as upward and downward movements and is visible in the heatmaps displaying the HHD-ray's hit points on the semi-transparent plane surrounding the (far) left (Fig. 4.9), (far) right (Fig. 4.10), (far) up (Fig. 4.7), and (far) down (Fig. 4.8) cubes.

These observations are also reflected in the way the device orientation affected the HHD-ray's accuracy (see Fig. 4.2). Compared to portrait mode, using the HHD in landscape mode lead to higher inaccuracies of the HHD-ray when focusing on cubes that are placed at the far left or far right position. On the contrary, using the HHD in portrait mode resulted in higher inaccuracies of the HHD-ray (compared to landscape mode) when focusing on cubes that are placed (far) up, (far) down, or (far) forward. These effects seem reasonable as the distance between the device center (i.e., the HHD-ray) and the display's left or right side is larger in landscape mode. Analogously, the distance between the display's center and its upper or lower side is larger in portrait mode. This effect was particularly strong for upward movements. This is plausible considering that these upward movements are physically more demanding as they are directed against gravity.

A comparison of the HHD-ray's accuracy between the devices, shows only slight differences except for the far left and far up cubes where the phone performed worse (see Fig. 4.4). This effect could be explained by the way the participants held the devices. While the tablet was mostly held with both hands, the phone was often only held with one hand.

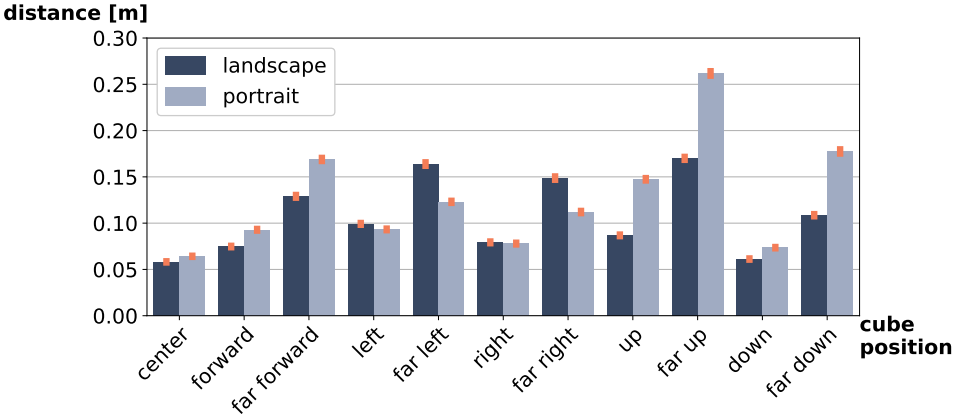


Figure 4.2: Mean distances (and 95% confidence intervals) between the HHD-ray's collision point on the semi-transparent plane and the cube's center using different device orientations.

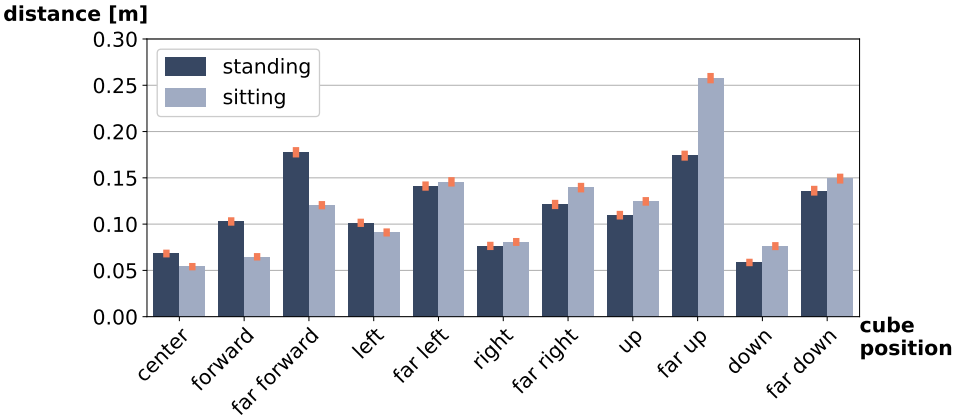


Figure 4.3: Mean distances (and 95% confidence intervals) between the HHD-ray's collision point on the semi-transparent plane and the cube's center in different body poses.

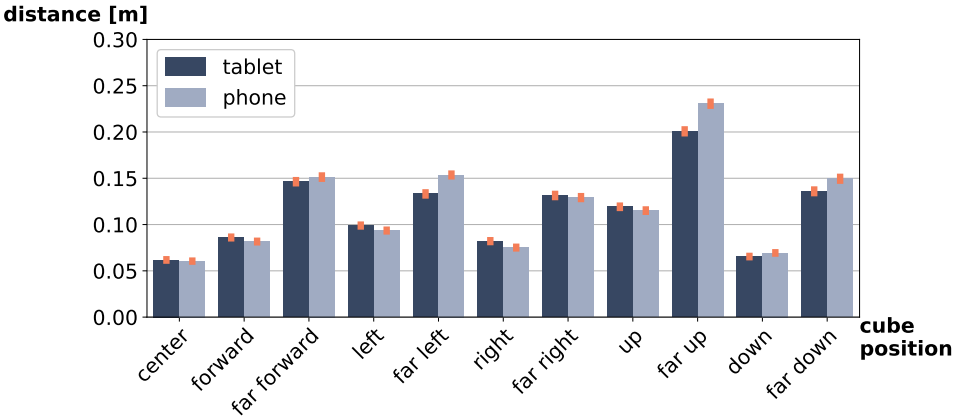


Figure 4.4: Mean distances (and 95% confidence intervals) between the HHD-ray's collision point on the semi-transparent plane and the cube's center with different devices.

This makes upward movements even more demanding and likely to stop as soon as the cube appears on the screen. Moving the HHD to the left is difficult if the device is held with the right hand only. If the phone is held with the left hand only, the front camera is likely being occluded in landscape mode where the front camera is located on the left display side. Since 95% of our participants' dominant hand was the right hand and it is difficult to hold the phone in landscape mode with the left hand without occluding the camera, movements to the left are likely to be stopped earlier too.

Furthermore, the HHD-ray's inaccuracies were higher while standing in the (far) forward task and for sitting in the far up task (see Fig. 4.3). An explanation for this effect could be the given by the lower device position while the participants were sitting. In this way, the device camera is likely to be exactly in front of the (far) forward cube such that the HHD-ray is more accurate. The heatmap displaying the HHD-ray's hit points on the plane surrounding the (far) forward cube (see Fig. 4.6) shows that the HHD-ray often hit the plane below the cube when participants were standing. We assume that this pattern was caused by the fact that while standing the participants held the HHD in a slightly tilted position such that the cube appeared at the upper part of the display and the HHD-ray hit the plane below the cube. On the contrary, the far up cube is further away while sitting. Thus, upward movements become even more demanding and device movement stops earlier.

Regarding face detection, we found that ARFaces were detected in more than 96% of all frames. Face detection performed worst for the far up cube (88%). For all the other cubes, face detection was successful in more than 95% of the frames. The majority of the frames in which face detection failed originated from tasks in landscape mode (79%) and while the participants were standing (62%). This appears plausible as in landscape mode the front camera is located at the left side of the display and thus more likely to be occluded by a hand and standing offers more possibilities for movement which could impede face detection.

At the end of the experiment, we asked the participants to rate the discomfort of each device $\times$ orientation configuration. The obtained ratings show that there was no clear preference for one of the configurations. Instead, it turned out that different participants preferred different device $\times$ orientation configurations. On average, the phone in portrait mode was rated most comfortable. This was followed by tablet in landscape mode. In this context, it is important to note that these are the most familiar configurations for both devices.

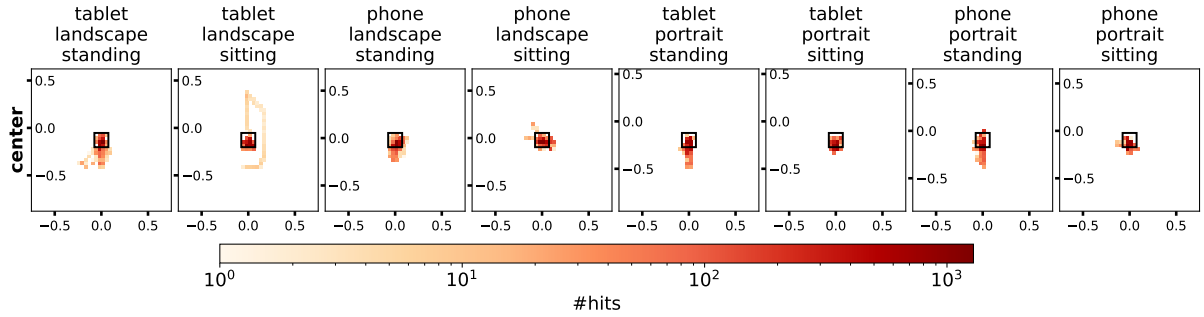


Figure 4.5: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the center cube.

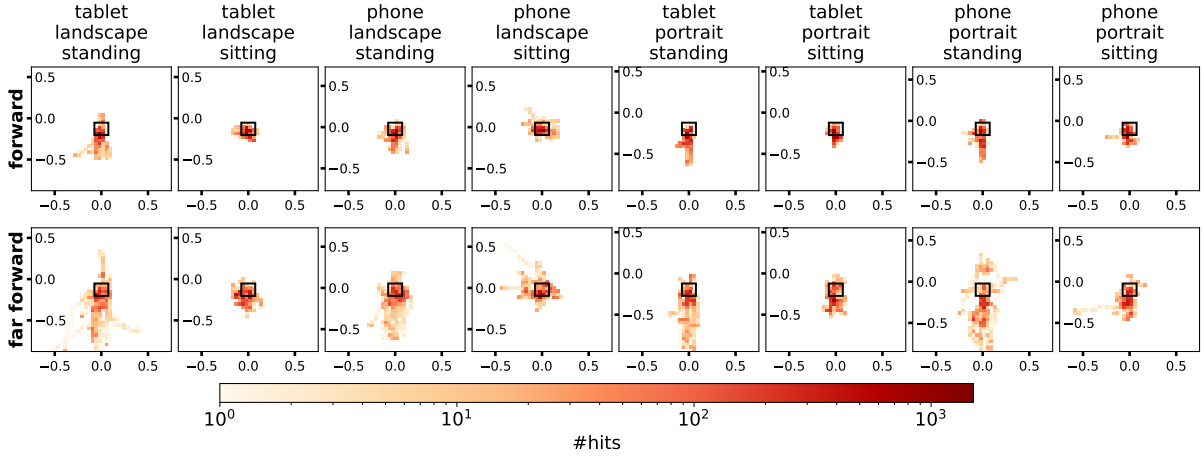


Figure 4.6: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the forward cube (first row) and far forward cube (second row).

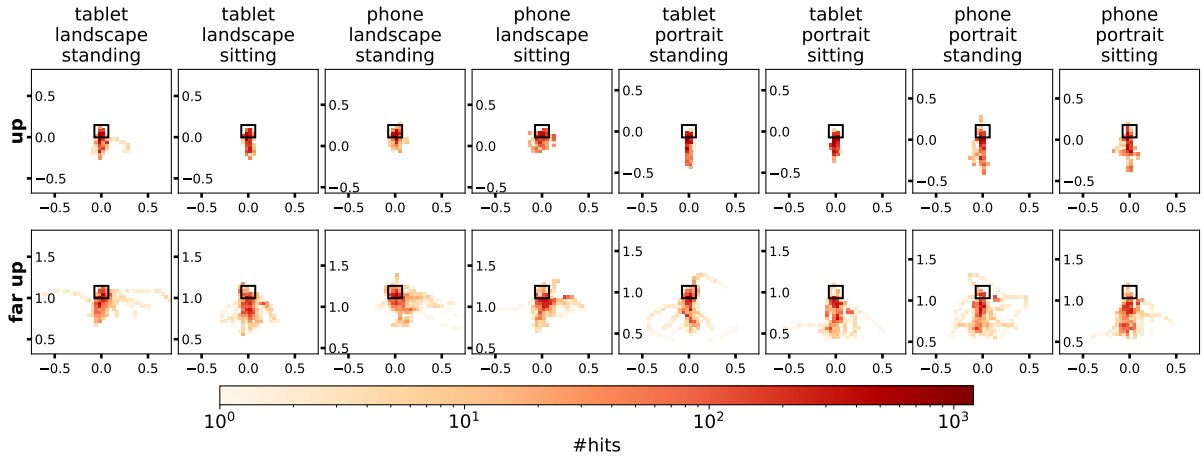


Figure 4.7: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the up cube (first row) and far up cube (second row).

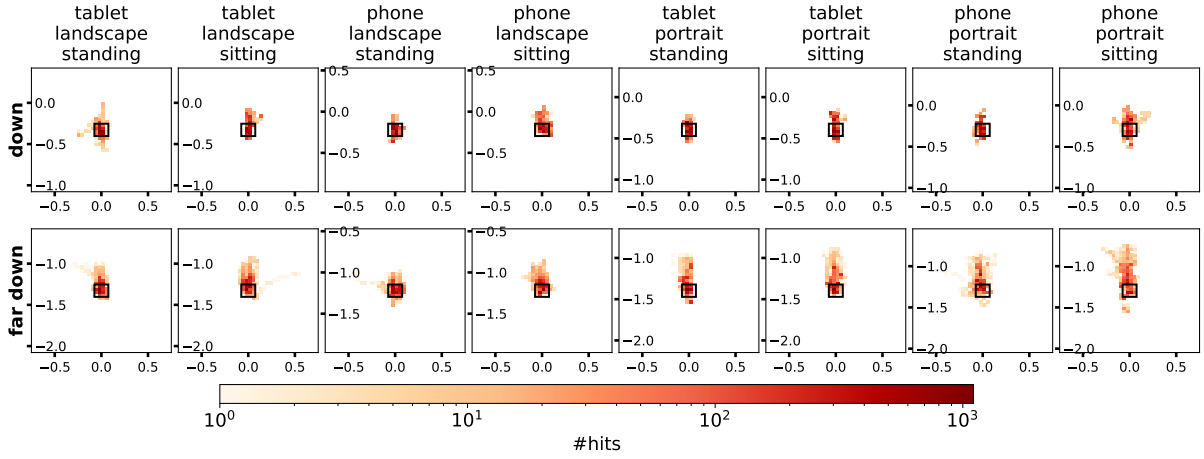


Figure 4.8: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the down cube (first row) and far down cube (second row).

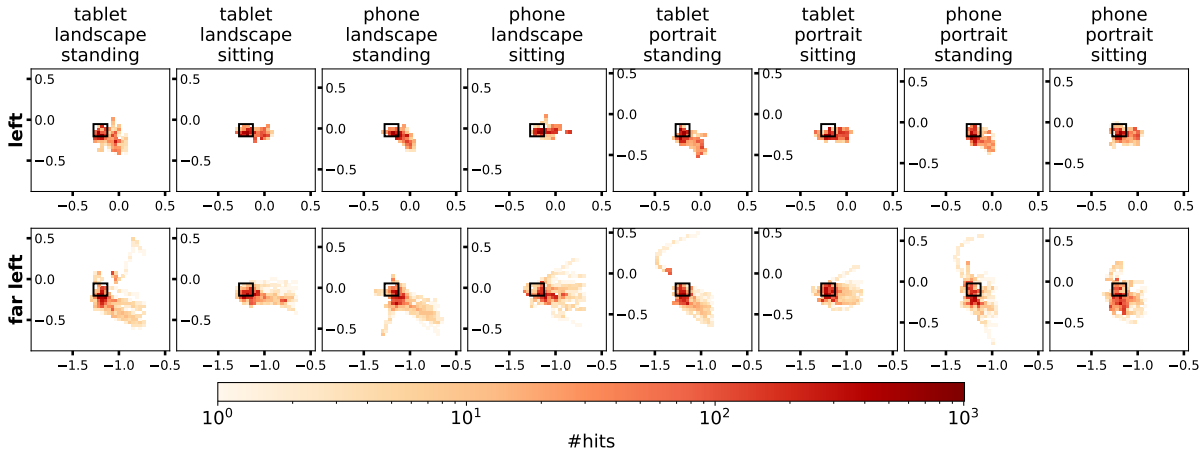


Figure 4.9: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the left cube (first row) and far left cube (second row).

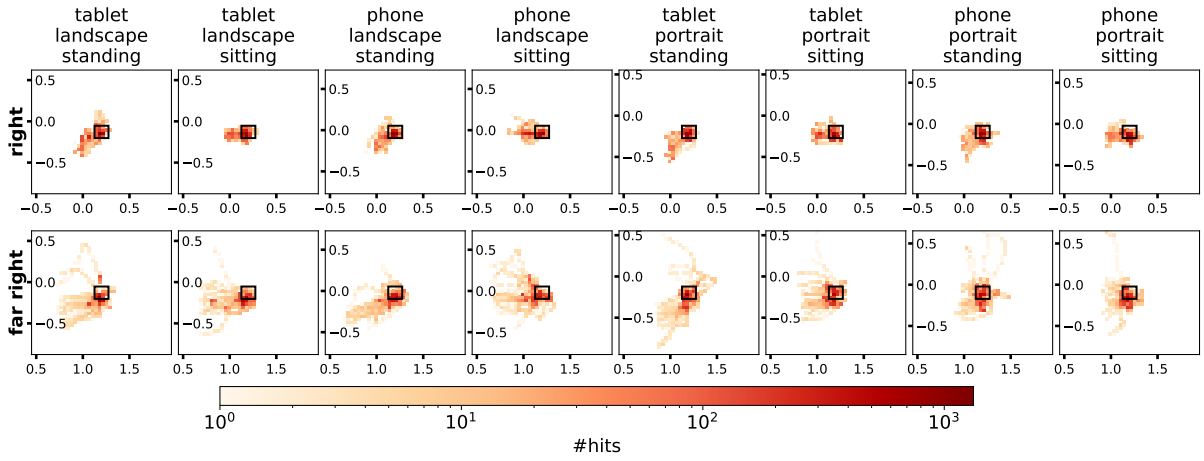


Figure 4.10: 2D (x,y) hit points of the HHD-ray on the semi-transparent plane surrounding the right cube (first row) and far right cube (second row).

Regarding the way the devices were held during the experiments, we observed that the tablet was mostly held with two hands in both landscape and portrait mode whereas the phone was held with one hand more often, especially in portrait mode.

4.4 Visual Cues for Diverse Extended Reality Technologies and Group Sizes

Apart from challenges arising due to the incorporation of different XR devices, the design of collaboration support features is also challenged by the number of participating collaborators. Addressing **Research Question 2** which concerns the avoidance of visual overload in large groups of participants, this section presents the design of mechanisms which allow individual activations of awareness cues.

4.4.1 Concept and Design

Previous research stated that the permanent activation of awareness cues should be avoided [71, 84] and raised the idea of adapting their visibility dynamically [139, 17]. Building on this, we designed four collaboration support features which allow individually activating awareness cues based on natural cooperation paradigms.

To this end, we consider the nine different collaboration styles presented in Fig. 2.1 and

design the mechanisms for synchronous co-located and distributed collaboration involving MR-HHDs, MR-HMDs, and VR-HMDs. As explained in Chapter 2.2, for pairs of different devices, two collaboration styles need to be considered. For example, the two collaboration styles *MR-HHD user sees VR-HMD user* and *VR-HMD user sees MR-HHD user* need to be handled differently as they pose different requirements to the visualization of the corresponding collaborator.

The designed mechanisms build up on a basic user representation which consists of avatars which are further equipped with view frustums, head-rays, and hand-rays. We decided to incorporate rays as we agree with Jing et al. [84], who note that rays can facilitate tracing the clue back to its origin. Based on this generic user representation we suggest adaptations for specific collaboration styles. Since co-located collaborators can naturally see each other, only distributed collaborators are provided with a visualization of each other's avatar. In co-located settings, we recommend the integration of an invisible avatar as this allows the collaborators to interact with each other like with distributed collaborators (e.g., pointing at the avatar's or person's body respectively).

Chapter 4.4 is based on those abstract avatar representations and does not further specify tracking approaches for collecting the underlying data. Considerations on this are outlined in Chapter 4.5 which also takes into account the results from Chapter 4.3.

The four mechanisms designed for individually activating awareness cues are called *GazeCollision*, *StareForCues*, *LookAtMe*, and *FreezeCues* (see Fig. 4.11). To describe their design, two exemplary collaborators *UserA* and *UserB* are used which serve as blueprints for all the nine collaboration styles.

On default, the head-rays and hand-rays of all collaborators are invisible for a user while a user's own rays remain visible to him or her. As soon as the collaborative session starts, the visibility of the awareness cues is then handled automatically by the four mechanisms. In this context it has to be noted that while the rays might not be visible, their position and orientation can be derived at all times to determine if a mechanism needs to be executed or not.

The first mechanism *GazeCollision* is triggered when UserA and UserB look towards the same area. This is the case when the collision points of their head-rays with scene components are within a specified distance for a specified time. *GazeCollision* then automatically activates the head-rays for UserA and UserB.

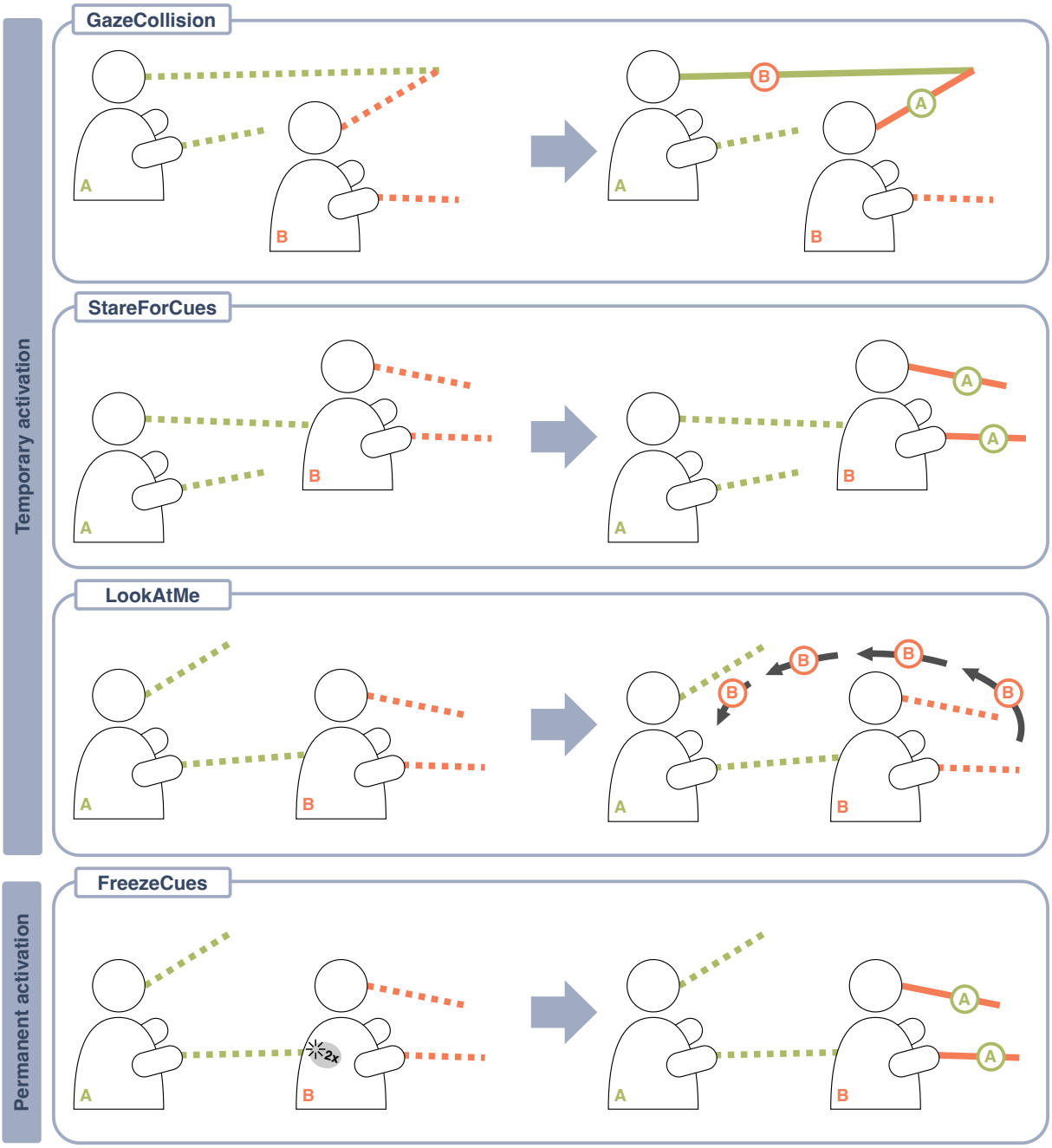


Figure 4.11: Temporary and permanent visibility of cues activated by *GazeCollision*, *StareForCues*, *LookAtMe*, and *FreezeCues*. Dotted lines indicate that the corresponding ray is only visible for its own user. Solid lines indicate that the ray is visible for another user: *GazeCollision* makes the head ray of UserA visible for UserB and vice versa, *StareForCues* makes UserB's head-ray and hand-ray visible for UserA, and *LookAtMe* adds arrows to the view of UserB. *FreezeCues* enables permanent activation of the head-ray and hand-ray.

If UserA wants to see the current activities of another UserB, *StareForCues* can be applied. By staring at UserB such that UserA's head-ray collides with UserB's body or avatar for a specified time, the head-rays and hand-rays of UserB become visible for UserA. As such, *StareForCues* takes up the corresponding face-to-face cooperation paradigm which also requires looking at someone to obtain information about this person's current activities. If UserB wishes to see UserA's activities as well, UserB can apply *StareForCues* too.

Apart from this, users can draw attention to themselves and take influence on another user's field of view. To this end, *LookAtMe* can be applied. To get UserB's attention, UserA can perform a single tapping gesture while the hand-ray collides with UserB's body or avatar. In this way, *LookAtMe* corresponds to the natural interaction paradigm of tapping on someone's shoulder. After *LookAtMe* has been performed by UserA, UserB's field of view is temporarily augmented with arrows pointing towards UserA. In a next step, UserA and UserB could then for example use *StareForCues* to activate each others' cues.

The arrows, head-rays and hand-rays activated by *GazeCollision*, *StareForCues*, and *LookAtMe* are disabled automatically after a specified time. The fourth mechanism *FreezeCues* can be applied at any time to activate the hand-rays and head-rays permanently. To this end, a double tap has to be performed while pointing at another user with the hand-ray. Thereby, tapping has to be implemented individually for the corresponding XR technology.

4.4.2 Considerations for Implementation

In the following we outline considerations for implementing the four mechanisms. Thereby, we distinguish between users and collaborators. The term *user* is used to describe this user's individual view to the XR scene including information about the user's *collaborators*. Each collaborator is considered an individual user as well.

Each user has a list of collaborators whereby each collaborator has a ⟨bool: co-located⟩ as well as three timers ⟨Timer: head_ray⟩, ⟨Timer: hand_ray⟩, and ⟨Timer: arrows⟩. The timers store a ⟨bool: permanent⟩ and a ⟨int: seconds⟩. This approach allows assigning different timers to different cues. At the start of a collaborative session, a list of all other collaborators is set along with ⟨bool: co-located⟩. Then, the initial scene is configured for each user. For each remote collaborator, the user's scene is augmented with a virtual avatar. Furthermore, invisible avatars are added for each co-located collaborator. In this

way, the user can interact with all collaborators (e.g., by pointing at their body or avatar).

For each user, the visibility of awareness cues (i.e., hand-rays, head-rays, and arrows) is then handled as illustrated in Fig. 4.12. The flowchart explains when the awareness cues describing the activities of UserB become visible for UserA. Considering Fig. 2.1, the flowchart summarizes the visibility of awareness cues for a single collaboration style (i.e., UserA sees UserB). If UserA is a MR-HHD user and UserB is a MR-HMD user, the collaboration style corresponds to the light blue arrow pointing from MR-HHD to MR-HMD at the bottom of Fig. 2.1.

Initially, all rays and arrows are disabled ■. Depending on the activities of UserA and UserB, UserA's field of view may then be adapted as follows. Starting from the dark-gray square on the left, three scenarios are possible. If UserA performs *StareForCues*, UserB's head-ray and hand-ray become visible for UserA. If instead, UserB performs *LookAtMe*, UserA's view is augmented with arrows pointing towards UserB. In the third scenario the head-rays of UserA and UserB collide such that UserB's head-ray becomes visible for UserA. In this case, UserA's head-ray will also be activated for UserB. However, since this adaptation concerns UserB's field of view it is handled outside the flowchart given in Fig. 4.12. Therefore, it will not be outlined further in this section. In any of the three scenarios, UserA's list of collaborators is searched for UserB and the respective timers of UserB are started. This also applies for the temporary cue activations described in the following.

In the flowchart, ellipses with white background represent cues which are only temporarily activated. In these cases, the visibility of the cues is disabled automatically as soon as the timer expired unless another event occurs prior to the expiration of the timer. At any time, UserA can permanently enable head-rays and hand-rays of UserB by applying *FreezeCues* (FC_on). To this end, UserA needs to perform a double tap which will set the corresponding boolean of the timers to true. To disable the cues and set these values to false again, another double tap (FC_off) has to be performed.

The arrows which appear upon the occurrence of *LookAtMe* cannot be enabled permanently and will disappear either automatically after the timer expired or after the occurrence of *GazeCollision*, *StareForCues*, or *FreezeCues*. As for example shown by the two ellipses on the right side of Fig. 4.12, UserA's field of view is temporarily augmented with arrows if UserB performs *LookAtMe* after the occurrence of *StareForCues* or after *GazeCollision*. When UserA performs FC_on, the arrows disappear and UserB's head-ray and hand-ray become permanently visible.

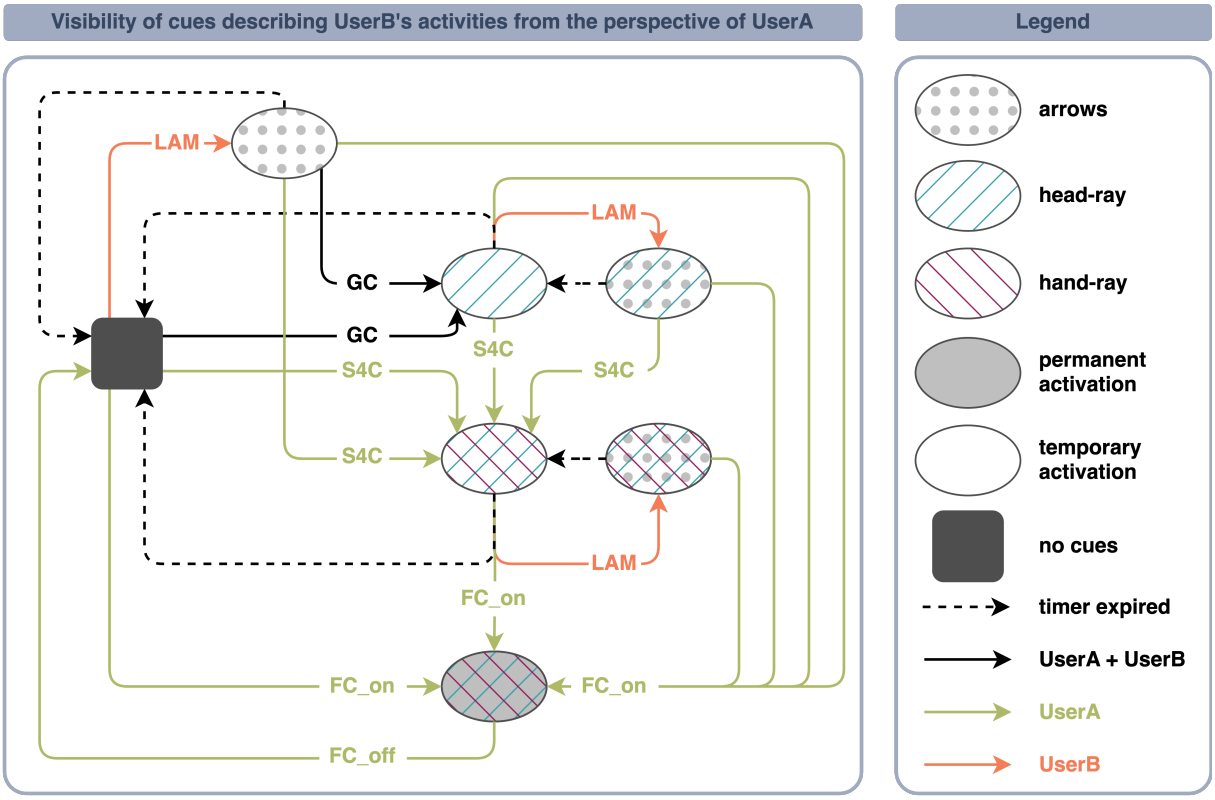


Figure 4.12: The visibility of cues describing the behavior of UserB from the perspective of UserA is adapted when a timer expires or based on the following activities of the users: UserA applies *StareForCues* (S4C) or *FreezeCues* (FC_on/FC_off); UserB applies *LookAtMe* (LAM); UserA and UserB meet through *GazeCollision* (GC).

When UserA performs *StareForCues* while only the head-ray and arrows are visible, the arrows disappear and UserB's head-ray and hand-ray become temporarily visible. If none of these actions are performed and the arrows' timer expired, the arrows will disappear automatically.

If UserB initially performs *LookAtMe*, the view of UserA is augmented with arrows pointing towards UserB. Upon the occurrence of *GazeCollision* the arrows disappear and the head-ray of UserB becomes temporarily visible for UserA. If instead UserA performs *StareForCues* or *FreezeCues*, the arrows disappear and both the head- and hand-rays become temporarily or permanently visible. If none of this happens prior to the expiration of the timer the arrows are disabled automatically.

If the cues of UserB were already permanently enabled by UserA and UserA has made an active decision to follow UserB's activities, *LookAtMe* performed by UserB will have no effect on UserA's view since these arrows are considered superfluous in most cases.

4.5 Discussion

Chapter 4 contributes to the design of collaboration support features for XR^S spaces which scale with varying XR technologies and group sizes. Providing all collaborators with the same set of awareness cues (**Objective 4**) such that each collaborator can see the location and orientation of all other collaborators (**Objective 1**) as well as their activities (**Objective 2**) can lead to incorrect user representations and visual clutter in cross-device settings and large groups.

While an HMD user's head orientation can be obtained from the head-worn device itself, the generation of corresponding user representations of HHD users is more complex since HHDs are not attached to the head. As a first step towards enhanced HHD user representations which prevent incorrect representations of user behavior (**Objective 3**), it was explored how accurately a ray originating from the center of an HHD determines the position of the virtual object the user is currently looking at in Chapter 4.3. In this context, we considered different types of HHDs which are used in different device orientations and body poses. In addition to evaluating the accuracy of the HHD-ray, we also assessed the accessibility of the front camera to explore its potential as a supplementary source for gathering information on user behavior, such as through face tracking.

During our investigations, different preferences regarding the combination of device (phone or tablet) $\times$ display orientation (landscape or portrait) were reported by our participants, reaffirming the importance of MR-HHD user representations which are applicable across different display sizes and device orientations. The need for such scalable solutions will further increase if a use case requires or excludes specific configurations. Developers of collaborative XR environments which include MR-HHDs should therefore consider different MR-HHD configurations and allow MR-HHD users to switch between them on-the-fly.

Overall, the results of our experiments indicate good performance of the HHD-ray's accuracy and face detection. The insights gained expand the options for collecting data about user behavior for generating suitable user representations. A more detailed discussion of these is provided at the end of this section.

Chapter 4.4 focuses on the avoidance of visual overload in collaborative settings of large groups. As the number of users in a collaborative XR setting increases, adding awareness cues which represent the activities of all users is prone to visual clutter which can cause visual overload, making it more difficult for each user to separate necessary from superflu-

ous information and match the visible awareness cues to the corresponding collaborator (**Objective 6**). To ensure that the visible awareness cues support the collaborative work process (**Objective 7**), the cognitive capacities needed for this should be kept as low as possible such that more capacities can be dedicated to the actual task.

Addressing this issue, we present mechanisms which individually configure the amount of visible awareness cues for each collaborator. Users can actively enable cues with *StareForCues* and *FreezeCues*. At the same time, they can draw attention to themselves with *LookAtMe* which allows them to activate cues in another collaborator's view. On top of that *GazeCollision* enables the visibility of awareness cues of collaborators who are focusing on the same area. To reduce cognitive efforts, the visibility of the cues is activated and deactivated based on natural cooperation paradigms such as looking at someone or tapping someone on the shoulder.

The presented mechanisms are based on two core cues: head-rays and hand-rays. The head-ray and hand-ray are naturally directed in similar directions since users rarely interact with or point at parts of a scene they are not looking at. This applies in particular to HHD settings where the hands are holding the HHD (i.e., the viewport to the MR scene). Still, the two rays display different aspects about the user's activities and are only meant to be enabled together if a user specifically requests so via *StareForCues* or *FreezeCues*. In order to make it easier for users to switch between the main access points of XR^S and allow them to intuitively control their own awareness cues (**Objective 5**) it is essential to apply the same mapping of user behavior to awareness cues for both HMDs and HHDs. Thus, if hand-rays of HMD users are generated based on hand movements, the hand-rays of HHD users should also be generated based on their hand movements. For HMD users, hand movements can be captured directly through hand tracking or indirectly by tracking controllers held in the hand. Furthermore, additional hardware such as wristbands could be attached to the user's arms or hands. Concerning hand-rays for HHD users, the HHD can be used as a laser pointer (i.e., allowing the user to continue holding the device with both hands).

Head-rays should ideally be generated by tracking the head orientation or eye movements of both HHD and HMD users. When choosing a tracking methodology, it is important to consider the differences between eye and head tracking. While rays displaying eye movements can potentially provide more accurate information about the user's viewing direction, they also change at a high frequency which can be considered uncomfortable for the other collaborators. On top of that, using the raw eye tracking data for the activation

of the mechanisms may lead to unintended actions as not all eye movements are performed consciously. To remove unconscious eye movement, data preprocessing must be performed. For HMDs the implementation of both tracking approaches is relatively straightforward whereas the generation of head-rays for HHD users is more complex.

The investigations on the front camera's accessibility described in Chapter 4.3 show promising results based on which we encourage the consideration of face tracking to generate head-rays under some constraints. In the experiments, the red borders, which appeared as soon as face detection failed, performed very well. Thus, we recommend maintaining a similar warning feature in future collaboration support features. In this way, the user can ensure the front camera's accessibility while looking at the screen. In our study, the front camera of both devices (tablet and phone) was located on the shorter edge of the display. As such, the front camera was located at the top when using the device in portrait mode and on the left side when using the device in landscape mode. In case an application does not require a specific device orientation, we recommend using devices (phone and tablet) with similar front camera location in portrait mode, as face detection failed less often when the HHD was used in portrait mode and the front camera is more likely to be occluded by a user's hand in landscape mode. Overall, face detection failed most often during upward movements. However, it is very unlikely that a user would move the device in this direction without looking on the screen as upward movements are considered physically demanding and uncomfortable. In these situations, integrating a virtual replica of the HHD to the remote collaborator's scene could prevent misunderstandings caused by false negative face detection. Alternatively, the visualization of awareness cues representing an HHD user's activities can be adapted when face detection fails, to indicate that the information delivered through the cues may be misleading.

Apart from this, the experiments revealed the HHD-ray to be a good proxy for the user's viewing direction. At the same time, we found that the HHD-ray's accuracy was affected by device orientation, viewing direction, and distance. Building up on the insights gained, an HHD user's viewing direction could also be visualized through an adaptive spotlight (instead of a ray) whose shape continuously adapts to the current device configuration and movements. Thereby, the device's orientation (landscape or portrait) and movement direction can be extracted during runtime. For instance, when the user performs movements to the left or right while the device is held in landscape mode, the shape of the spotlight should reflect inaccuracies along the x-axis (i.e., the spotlight should be flat and wide) while upward or downward movements performed while holding the device in portrait mode would require a long and narrow spotlight. The area of collisions evoking

mechanisms such as *GazeCollision* or *StareForCues* can thereby be extended in line with the shape of the spotlight.

Thus, the approach for generating the head-ray of HHD users may be chosen depending on use case specific constraints, user preferences, and the accuracy of available face tracking solutions. The accuracy of the HHD user's head-ray may be further improved by combining information on the viewing direction from face tracking and adaptive spotlights.

Parts of this chapter have been previously published in:

V. M. Memmesheimer, S. M. Schwenkreis, and A. Ebert (2024): Towards Enhanced User Representations for Handheld Mixed Reality. In Mensch und Computer 2024 – Workshopband. Gesellschaft für Informatik e.V. doi: 10.18420/muc2024-mci-ws06-205.

V. M. Memmesheimer, J. Löber, and A. Ebert (2024): AWARE^{S}CUES: Awareness Cues Scaling with Group Size and Extended Reality Devices. In J. Y. C. Chen and G. Fragomeni (Eds.) Virtual, Augmented and Mixed Reality. HCII 2024. Lecture Notes in Computer Science, vol. 14706, pp. 44–59. Springer, Cham. doi: 10.1007/978-3-031-61041-7_4.

Chapter 5

Scalable Interaction Techniques for Extended Reality

The previous chapter contributes to collaboration support features which provide enhanced scalability along all dimensions of XR^S (i.e., varying degrees of virtuality, devices, and group sizes). In this chapter, we enhance the scalability of interaction techniques – another essential aspect of XR^S. More specifically, we develop and evaluate an interaction paradigm for object manipulation which scales across different degrees of virtuality and devices. The proposed paradigm applies to all access points of XR^S, allowing both individual users and collaborators to seamlessly switch between them.

The chapter begins with an overview on research and aspects related to the design of our novel interaction paradigm. As such, Chapter 5.1 summarizes previous work which either focuses on device-based interaction methods for MR-HHDs or combines HMDs with HHD-based controllers. Based on the insights gained from previous research and the requirements of the XR^S framework, we then determine objectives for the design of a scalable interaction method in Chapter 5.2 and translate them into two research questions. The first research question concerns the design of an object manipulation paradigm for MR-HHDs which entails additional challenges due to the HHD's dual role as input and output modality. To facilitate switching between access points of XR^S, the second research question concerns the extension of the resulting MR-HHD interaction technique to HMDs.

Addressing the first research question, Chapter 5.3 presents the design and implementation of a novel device-based interaction paradigm for MR-HHDs called *Move'n'Hold* along with an initial evaluation. Chapter 5.4 then demonstrates how MR-HMDs and VR-HMDs can be combined with a tablet controller which implements the same interaction

paradigm as *Move'n'Hold* for MR-HHDs. In addition, it reports the results of a second, detailed user study in which the set of interaction techniques provided by *Move'n'Hold* for MR-HHDs, MR-HMDs, and VR-HMDs is evaluated against a set of state-of-the-art techniques. Eventually, Chapter 5.5 provides a joint discussion of the developed interaction paradigm and results of both evaluations.

5.1 Related Research and Aspects

HHDs such as tablets and smartphones are ubiquitous in today's society. Thus, MR-HHDs provide a highly accessible and low-cost entry point to XR^S. Although (multi) touch gestures are well-established as an intuitive way to interact with HHDs in non-XR settings, they do not transfer well to spatial interaction.

In MR applications, HHDs act as windows to the MR scene and therefore need to be held up higher than in non-XR settings. Furthermore, the interaction with and manipulation of virtual content which is anchored in 3D space requires options for spatial input. These differences pose new challenges to the development of interaction methods for MR-HHDs.

As described in Chapter 3.1.4, touch-based and gesture-based interaction techniques are deemed unfavorable for interacting with MR-HHDs as they require the HHD to be held with one hand and are thus prone to fatigue. At the same time, the hand touching the screen or performing the gesture is likely to occlude the MR scene. Here, a promising alternative is offered by device-based interaction techniques [62]. This approach maps the HHD's movement to the virtual objects. As such, it supports spatial input while allowing users to hold the HHD with both hands. In the following, a short overview of previously proposed device-based interaction methods is given.

Object manipulation commonly starts with the selection of the object. To this end, some approaches combined device-based manipulation with touch-based selection (e.g., [120]). However, the combination of touch-based selection and device-based manipulation requires frequent adjustments of the hands' poses due to temporary one-handed interaction during selection. Other approaches allow selecting objects by centering them on the screen (e.g., [15, 161]), which is similar to gaze-based selection with HMDs and enables HHD users to always hold the device with both hands.

Upon selection, the object can be manipulated as device movements are mapped to the

virtual object. For example, Marzo et al.'s [109] device-based manipulation method maps device movements to the object whereby the object's position and orientation relative to the device stay the same. In this way, the object does not move out of view during rotations. The method implemented by Grandi et al. [68] maps device movements to the object as long as a peripheral button on the right side is pressed. Similar to [109], the object thereby stays at a constant position and orientation relative to the HHD. In the approach presented by Blattgerste et al. [15] a grabbing metaphor was used. As such, the object is automatically placed in front of a smartphone and can be manipulated by moving the device into the desired position and orientation. In contrast to [15, 68, 109], the UI proposed by Mossel et al. [120] allows performing translation and rotation separately by mapping the HHD's pose to the objects accordingly. Alternatively, objects can also be translated and rotated simultaneously. Thereby, rotations are performed around the point at which the raycast triggered via touch input hit the object.

Samini and Palmerius [149] point out that objects attached to the device are hard to rotate without translating them, and objects that are rotated relatively around their own center will move out of view during rotations around the x- and y-axis. This issue was also mentioned by Mossel et al. [120]. Addressing this topic, Samini and Palmerius [149] proposed an approach that allows rotating objects relative to the device while user perspective rendering is implemented to prevent the object from moving out of view. However, this approach still requires repositioning the device during large rotations. This manner of interaction is referred to as *clutching* which is needed when an object cannot be manipulated in one motion but instead must be released between successive manipulations [194, 18]. In this context, Marzo et al. [109] note that during large rotations performed with the device-based approach, the task was split to consecutive movements and slowed down task completion. Clutching was also used in the method by Grandi et al. [68]. In some studies device-based interaction showed promising results in comparison to touch-based interaction for manipulation tasks which involve combinations of translation and rotation [120], device-based rotation (especially large rotations), however, remained difficult [109, 68].

Addressing this issue, Su et al. [161] introduced an object rotation technique that continuously rotates objects around a constant speed when the HHD's rotation exceeds predefined threshold values. This rotation method is integrated in a system which allows combining translation and rotation for device-based object manipulation. Objects can be selected through raycasting from the screen's center and then translated or rotated separately as long as a peripheral button is pressed. Thereby, continuous manipulation is only available for large rotations while large translations still have to be performed by repositioning the

device. The authors report positive results regarding continuous movement as well as the separation of translation and rotation. Their approach, however, limits flexibility and controllability for the user due to predefined thresholds and speed. Furthermore, users are required to reposition the device for starting and resuming continuous movement.

Standard interaction techniques for HMDs which rely on mid-air gestures and controllers are often physically demanding, imprecise, or require external tracking systems. As a promising alternative, previous work has considered the integration of mobile devices like phones or tablets which provide highly accessible and advanced input options while offering a secondary display. In this context, researchers have explored the integration of smartphones [89, 94, 143, 153, 174, 196] and tablets [77, 98, 104, 162] with VR-HMDs [89, 98, 153, 162, 195] as well as with MR-HMDs [77, 94, 104, 143, 174, 196] for various purposes. An overview of these approaches is presented in the following.

In the context of HMDs, Knierim et al. [94] combined a MR-HMD with a phone which allows manipulating objects through different touch gestures. Whenever the user begins to translate an object, a reference coordinate system is set up. Objects can then be translated horizontally by single taps followed by a swipe. A double tap followed by a swipe enables vertical translations. Rotations around the y-axis can be performed with multi-touch (i.e., two finger rotations on the screen). Compared to mid-air gestures, this approach enhanced accuracy, and reduced task completion time as well as taskload. Furthermore, the authors mention that the HHD-based approach allows manipulating objects in more relaxed postures. Another approach of a MR-HMD extended with a phone was proposed by Unlu and Xiao [174]. Here, the phone acts as a 6 DOF input device as well as a 3D trackpad. A virtual plane is attached to the phone such that virtual objects that are located on this plane can be moved on the plane through touch gestures on the phone's display. Luo et al. [104] attach the virtual object seen though a MR-HMD to a tablet. The tablet's movement in space is then mapped to the object. Thus, the object's position and orientation are manipulated and updated at the same time. Thereby, clutching could be used to enhance comfort and reduce physical effort.

Kari and Holz [89] connected a phone to a VR-HMD which can be used to adjust the position and orientation of two virtual hands. Both hands are located on a plane which is generated according to the phone's position and orientation and constantly adapted based on the phone's movement. The user can control the hands through touch input applied with the thumbs on the display's left and right side. Large hand movements are enabled through clutch during touch input and phone movement amplifications for computing the

plane position. Objects can be picked by hovering the desired object with the virtual hand and then applying touch on the phone's display. The object is placed when touch is released. Other approaches involving VR-HMDs have been proposed by Surale et al. [162] who enabled object manipulation in VR through a tablet as well as by Zhao et al. [195] who extended the VR-HMD with a custom, tablet-like device that is able to provide haptic feedback. Both approaches integrate a visualization of the tablet in the VR scene. In [195], the complete VR scene is rendered on the virtual tablet and frozen upon the initiation of object manipulation. Thus, objects can be selected by touching them on the tablet's screen and translated and rotated through touch gestures. In [162], the tablet acts as a viewport. For object selection, users can point at the desired object with a ray emerging from the tablet and then tap on its screen. Alternatively, the object can be selected by touching the virtual object with the physical tablet or touching it directly with the hand. Selected objects can then be manipulated through touch gestures in 9 DOF (i.e., translating, rotating, and scaling objects). Thereby, different tablet orientations can be used to fix an axis.

5.2 Objectives

As summarized in Chapter 3.1.4 and Chapter 5.1, extensive research has focused on improving the usability of interaction techniques for MR-HHDs, MR-HMDs, or VR-HMDs. On the contrary, an interaction paradigm which is applicable to different XR technologies and enables seamless switching between them is still missing. However, seamless switching is a key requirement of the XR^S concept.

Thus, the design of a novel spatial interaction paradigm should address the following requirements from Chapter 3.3.2.

- **REQ 17** Users can intuitively switch between devices.
- **REQ 18** Users can intuitively switch between degrees of virtuality.
- **REQ 20** The interaction techniques for manipulating and referencing real and virtual components provide high usability.

The requirements **REQ 17** and **REQ 18** can be specified further in the following objec-

tives for scalable spatial interaction techniques:

- **Objective 1** All access points (devices) provide options for the same operations.
- **Objective 2** The same operation can be performed with the same interaction paradigm on all devices.
- **Objective 3** Using the interaction paradigm on one device serves as a training phase for using it on another device (cross-device benefits).

The novel interaction technique presented in this dissertation is focused on two types of spatial interaction: object translation and object rotation. While previous research has proposed translation and rotation methods for all access points of XR^S (**Objective 1**), the proposed methods rely on different interaction paradigms (**Objective 2**). Instead of transferring well-established interaction paradigms for 2D interaction to 3D, effective spatial interaction may require a completely new interaction paradigm. If such a novel interaction paradigm is specifically tailored to spatial interaction and stays consistent across different technologies, the initial effort to learn this technique may be compensated over time and should eventually result in cross-device benefits (**Objective 3**).

Apart from consistency across XR technologies, the novel interaction paradigm should also provide high usability (**REQ 20**). Here, we focus on interaction techniques for object manipulation. Hence, based on the lessons learned from previous research in Chapter 5.1 we specify **REQ 20** further through the following objectives.

- **Objective 4** The interaction paradigm separates object translation and object rotation.
- **Objective 5** The interaction paradigm allows holding HHDs with two hands throughout interaction.
- **Objective 6** The interaction paradigm supports spatial input.
- **Objective 7** The interaction paradigm applies to object translation and object rotation.

- **Objective 8** The interaction paradigm provides scalability with respect to different directions, distances, and complexities while keeping physical movement and cognitive efforts as low as possible.

- **Objective 9** The interaction paradigm provides high controllability.

Object translation and rotation involves 6 DOF. The optimal number of DOFs which are manipulated simultaneously does not only depend on the interaction technique but is also influenced by the task objectives. For example, DOF separation can enhance precision but also increase task completion time [113]. Yet, there is general agreement on the recommendation to separate translation and rotation [108, 113, 114] (**Objective 4**).

A key learning concerning spatial interaction with MR-HHDs is that devices should be held with both hands to minimize fatigue and occlusion (**Objective 5**). For HMDs which integrate input through an HHD controller, the manner in which the HHD controller is held is less critical as it can potentially be held in a lower, less fatigue-prone position. Although touch-based controller-operation (e.g., [94, 195]) has shown promise for HMDs, these interaction paradigms conflict with the principle of two-handed interaction required for HHDs.

Device-based interaction does not only allow holding the device with both hands but also offers spatial input (**Objective 6**) which allows translating or rotating the device to manipulate the object accordingly. For device-based interaction with MR-HHDs, a critical aspect concerns their dual role as input and output modality. To ensure the visibility of the manipulated object during device movement, large changes in device orientation should be avoided. Thus, object manipulation approaches for HMDs which use the HHD as a 3D trackpad such as [174] are not transferrable to HHDs.

To keep efforts as low as possible when users have to toggle between translation and rotation mode, the same interaction paradigm should be available for both translation and rotation (**Objective 7**). While mapping device movements directly to objects meets this objective for small manipulations, there is no consistent approach for handling large manipulations. To perform large translations the user may have to walk with the device whereas large rotations, especially around the x-axis, are complicated by the user's wrist movement limitations. In this context, previous work has implemented amplifications of phone translation [89] and considered clutching [109, 68, 104, 89]. To prevent users from having to reposition the device, clutching, however, should be reduced whenever

possible [18]. In the context of rotation, previous work therefore suggested the integration of automated continuous rotations [161].

Thus, a joint interaction paradigm which scales in terms of directions, distances, and complexities (**Objective 8**) and provides high controllability (i.e., users can individually activate continuous movement, adjust its speed and switch between small precise and large coarse movements) (**Objective 9**) is still missing.

Since object manipulation with MR-HHDs entails more restrictions with respect to the interaction design, we first developed an interaction paradigm for MR-HHDs and then extended this paradigm to MR-HMDs and VR-HMDs. More specifically, this chapter thus addresses the following research questions:

- **Research Question 1:** How can an object manipulation paradigm for MR-HHDs be designed which addresses objectives 4 – 9?
- **Research Question 2:** How can such an object manipulation paradigm be extended for HMDs, meeting objectives 1 – 3?

5.3 Move'n'Hold for Handheld Displays

Addressing **Research Question 1** this chapter presents the design, implementation, and evaluation of *Move'n'Hold* – a highly scalable interaction paradigm which allows translating or rotating virtual objects seen through a MR-HHD.

5.3.1 Design

Move'n'Hold builds on device-based interaction methods which map device movement to virtual objects (i.e., in the following referred to as *Move*) and extends this natural object manipulation paradigm with automated continuous movement; in the following referred to as *Hold*. The object manipulation process with *Move'n'Hold* can be divided in four steps (see Fig. 5.1) and is designed as follows.

To translate or rotate an object, a user can ❶ specify the desired object by moving the HHD such that the desired object appears at the center of the screen. As soon as the

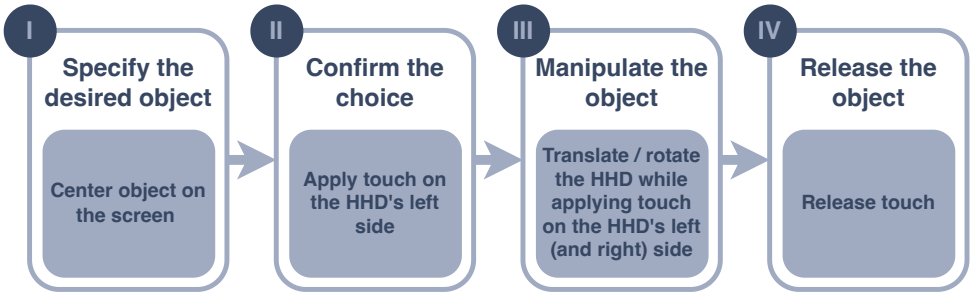


Figure 5.1: Object manipulation steps using *Move'n'Hold*.

object is hit by an invisible ray which emerges from the device center, the object changes its color as visual feedback (see Fig. 5.2). The user can then II confirm the selection of this object by applying left-thumb-touch. Touch can be applied anywhere on the left side of the screen such that the device can still be held with both hands.

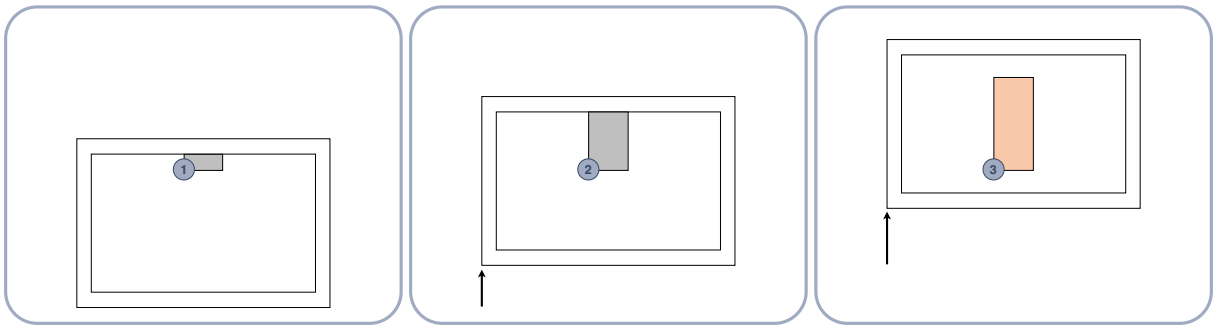
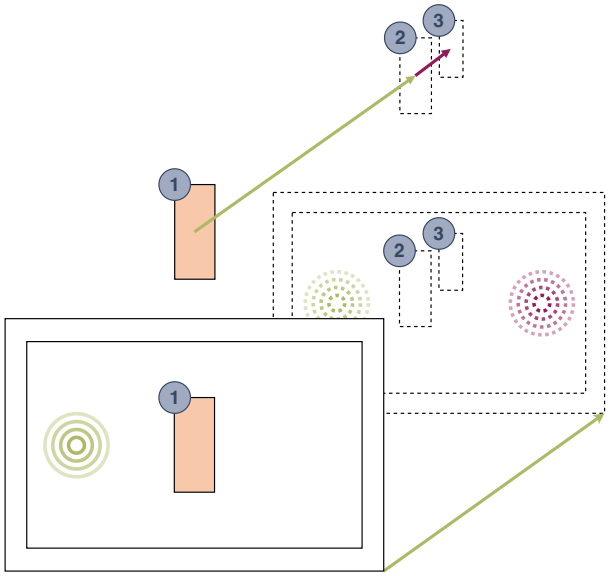


Figure 5.2: To select an object a user has to move the HHD such that the desired object is hit by an invisible ray emerging from the device center.

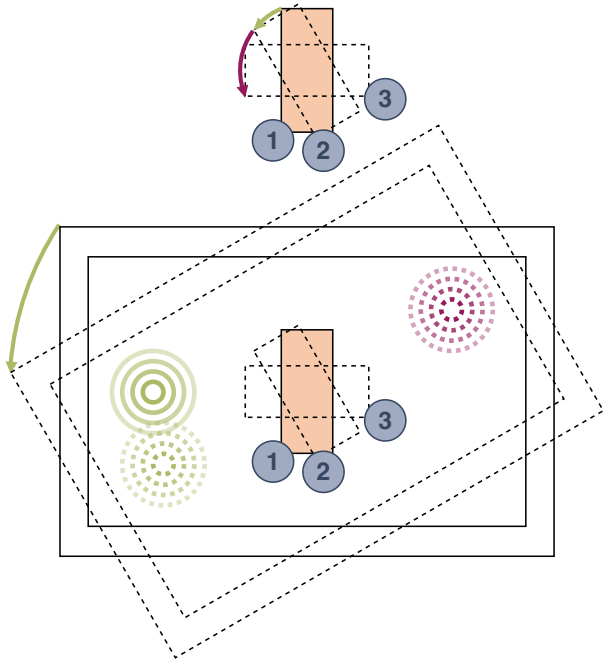
Depending on the active manipulation mode, the user can then III translate or rotate the selected object by moving the device while applying peripheral touch. Thereby, natural manipulation through direct mapping and automated continuous movement can be combined as illustrated in Fig. 5.3.

As long as only left-thumb-touch is active, the HHD's translation/rotation in space is mapped to the object relative to the object's center (i.e., manipulation with *Move*). Thereby, the HHD can be moved in three dimensions to either translate or rotate the object along or around multiple axes at once. In Fig. 5.3, this initial movement is illustrated by the green arrow.

To enable large manipulations we extend this basic manipulation method with Hold, an option for automated continuous manipulation. To evoke continuous manipulation after an initial object manipulation with *Move*, the user can add right-thumb-touch while the left thumb remains on the screen. Again, touch can be applied anywhere on



(a) Object Translation



(b) Object Rotation

Figure 5.3: A MR-HHD user can translate/rotate a virtual object (here: a virtual rectangle (1)) as follows: First, left-thumb-touch has to be applied while translating/rotating the device. As long as touch is registered on the left side, the device's translation/rotation in space is mapped to the rectangle such that it (2) follows the movement of the device. By adding right-thumb-touch, the rectangle can be (3) translated/rotated automatically (i.e., without further device movement) in the direction of the initial movement. The automated movement of the rectangle stops as touch is released on the right side.

the right side of the screen. In this way, the object is translated/rotated automatically (i.e., without moving the HHD) in the direction specified during the initial movement. In Fig. 5.3, this continuous translation/rotation is illustrated by the purple arrow which is directed in the same direction as the green arrow. This manipulation of the selected object continues automatically as long as touch is registered on both sides. The speed of the automated movement can thereby be set individually by adjusting the length of the initial movement.

When touch is registered on both sides, the user can proceed in different ways. By IV releasing touch on both sides manipulation is stopped and the continuous manipulation direction is reset. Alternatively, the user can release touch only on the right side while maintaining left-thumb-touch to either resume object manipulation via direct mapping or to stop and start continuous movement by repeatedly applying and releasing right-thumb-touch while keeping left-thumb-touch active. All of these options are available for both object translation and rotation.

We found that performing one dimensional object manipulations, especially rotating an HHD precisely around a single axis, can be challenging. To investigate the usefulness of axis locking, a toggle button is provided during object rotations which enables the user to lock axes and rotate the objects only around a single axis. As illustrated in Fig. 5.5, the user can repeatedly click a button to toggle between rotations around all axes, and rotations only around the x-, y, or z-axis.

In contrast to Su et al. [161] whose object manipulation method includes continuous movement based on predefined thresholds and constant speed only for rotating objects, *Move'n'Hold* is applicable for both rotation and translation and allows starting continuous movements at any time and speed as well as stepwise manipulations without repositioning the device.

Since input is solely provided through the HHD's movement and peripheral touch on the sides of the display which can be performed without having to adjust the hand's positions, *Move* and *Move'n'Hold* allow users to translate or rotate objects while holding the HHD with both hands, with *Move'n'Hold* requiring less user movement (see Fig. 5.4).

5.3.2 Implementation

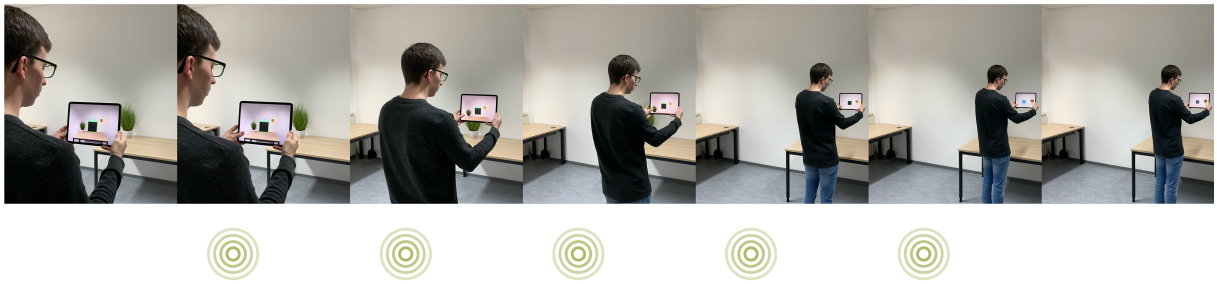
The design of *Move'n'Hold* was implemented as an MR-HHD application running on an Apple iPad Pro (11 inch, Gen. 3). It was developed with Apple's ARKit through Unity's AR Foundation package and deployed via XCode. The coordinate system's origin corresponds to the HHD camera's position and orientation when the application is launched.

As long as no touch input is registered on the tablet's screen, the application checks for collisions between a ray shot from the device camera's center and manipulable objects. When centering a manipulable object on the screen, its collider will be hit by the ray and the object will be highlighted by changing its color to green. The object is then saved as the current target object and the user can start manipulating the object by combining device movements with peripheral touch input. While manipulation is in progress (i.e., at least left-thumb-touch is active), ray shooting pauses and continues when no touch is registered. Depending on the active manipulation mode, the selected object is then manipulated as described in the following.

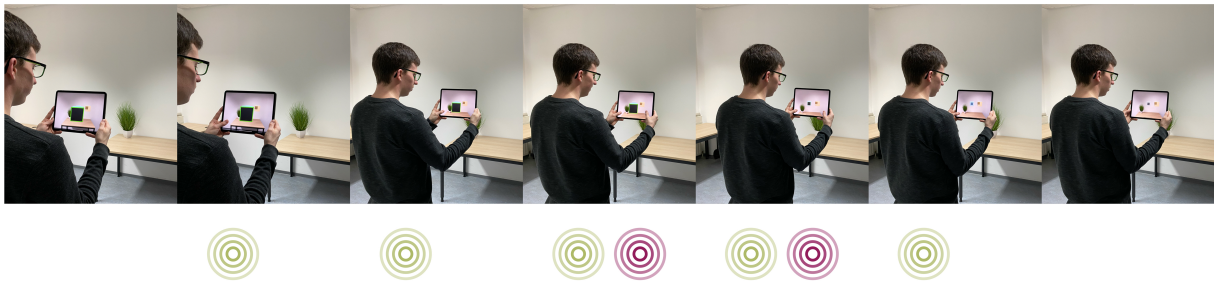
In translation mode, we update the selected object's position as long as left-thumb-touch is active through direct mapping. In each frame, the vector v_{move} describing the HHD's translation between the previous and the current frame is computed and added to the object's position v_{obj} (Eq. 5.1). To this end, the HHD's position is registered and stored in each frame.

When touch is registered on the left and right side of the display, we first compute the vector v_{hold} (Eq. 5.3) which describes the HHD's translation while only left-thumb-touch was applied (i.e., the translation from v_{ltt} when left-thumb-touch was first registered to v_{rtt} when right-thumb-touch was first registered). While touch is registered on both sides of the display, the object will be translated in the direction of this vector.

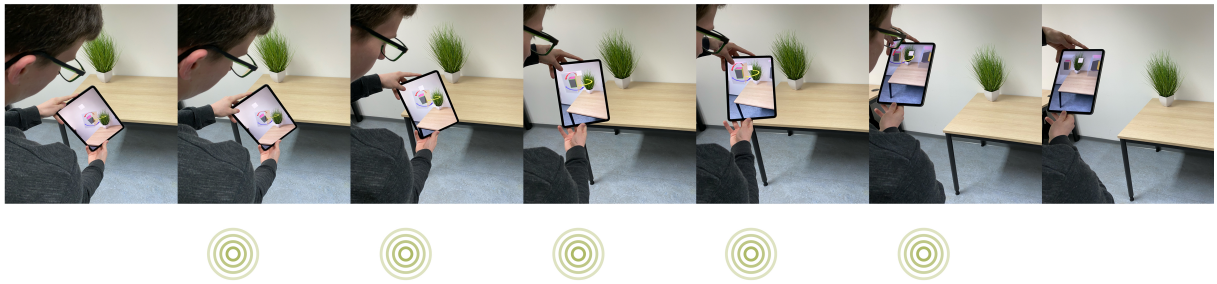
To prevent very fast movements and ensure that the user remains in control of object manipulation, we perform a linear interpolation by 0.1 as the interpolant which was determined to provide good control in a pre-study. More specifically, we add v_{hold} to the object's current position v_{obj} and linearly interpolate with 0.1 between the object's current position v_{obj} and the obtained value (Eq. 5.5). Thereafter, the object's position is updated to the interpolated value v_{lerp}. This procedure is repeated in every frame as long as right-thumb-touch remains active. In this way, the object moves automatically (i.e., independent of the device movement). The vector v_{hold} is stored until touch is released on both sides.



(a) Object Translation using *Move*



(b) Object Translation using *Move'n'Hold*



(c) Object Rotation using *Move*

(d) Object Rotation using *Move'n'Hold*

Figure 5.4: The photoseries show how a translation (a+b) and rotation (c+d) task is performed with *Move* (a+c) and *Move'n'Hold* (b+d). A comparison of the photoseries shows that *Move'n'Hold* requires less user movement than *Move*.

$$v_{\text{objNewDM}} = v_{\text{move}} + v_{\text{obj}} \tag{5.1}$$

$$q_{\text{objNewDM}} = q_{\text{move}} * q_{\text{obj}} \tag{5.2}$$

$$v_{\text{hold}} = v_{\text{rtt}} - v_{\text{ltt}} \tag{5.3}$$

$$q_{\text{hold}} = q_{\text{rtt}} * q_{\text{ltt}}^{-1} \tag{5.4}$$

$$v_{\text{objNewCM}} = v_{\text{lerp}} = Vector3.Lerp(v_{\text{obj}}, v_{\text{hold}} + v_{\text{obj}}, 0.1) \tag{5.5}$$

$$q_{\text{objNewCM}} = q_{\text{lerp}} = Quaternion.Lerp(q_{\text{obj}}, q_{\text{hold}} * q_{\text{obj}}, 0.1) \tag{5.6}$$

Object rotation is realized with the same interaction paradigm. Instead of the HHD's position, the rotation of the HHD is used to rotate objects.

If an object is selected and touch is registered on the left display side, the object's current orientation in space q_{ltt} is stored. As long as left-thumb-touch remains active, the object's orientation is then updated according to the device's rotation. To this end, the quaternion q_{move} which describes the device's rotation in the last frame is multiplied by the object's current orientation q_{obj} in every frame (Eq. 5.2). Analogously to object translation, the HHD's orientation is therefore saved in every frame.

Upon the registration of right-thumb-touch, we compute and store the quaternion q_{hold} (Eq. 5.4) which describes the HHD's rotation from q_{ltt} to the device's current orientation q_{rtt} for the continuous rotation. As long as touch is registered on both display sides, the object is then rotated automatically. Therefore, we multiply q_{hold} by the object's current orientation q_{obj} and perform a linear interpolation with 0.1. In every frame, the object's orientation is then updated to the interpolated value q_{lerp} (Eq. 5.6) as long as right-thumb-touch remains active. Again q_{hold} is stored until touch is released on both sides.

When the user limits rotation to a specific axis (see Fig. 5.5), rotations around the other axes are removed before mapping the device's rotation to the object and the visibility of the auxiliary arrows is adapted accordingly.

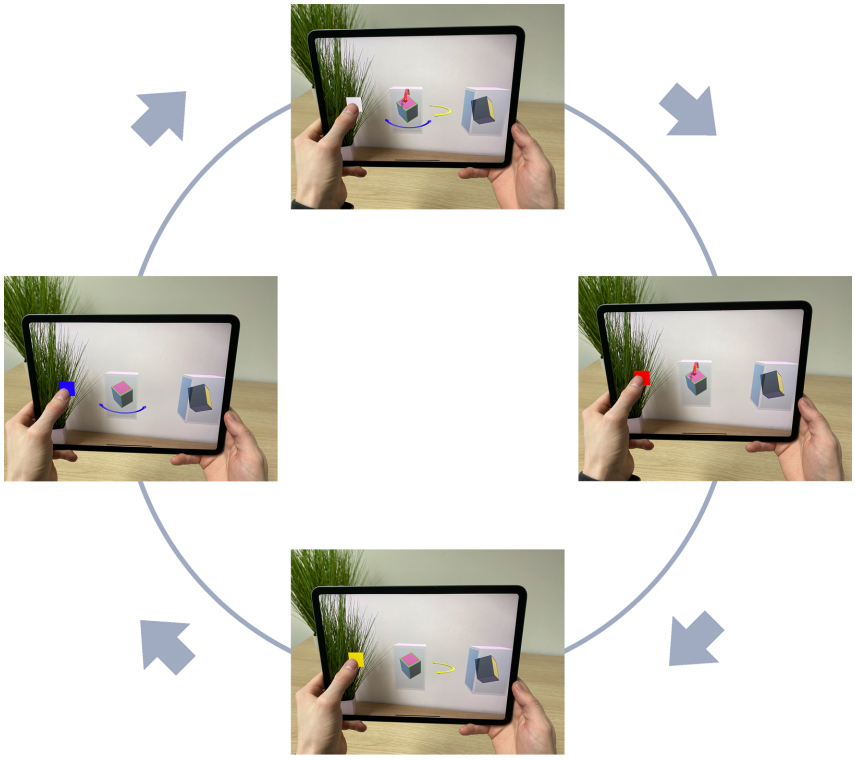


Figure 5.5: During rotation tasks, specific axes can be locked such that only the HHD's rotation around the selected axis is mapped to the object being rotated. Device rotations around other axes are ignored. By pressing the peripheral button on the left display side, the user can toggle between four options: no axis locking (white), only x-axis rotations (red), only y-axis rotations (yellow), only z-axis rotations (blue). The button's color and the auxiliary arrows are adapted depending on the user's choice.

5.3.3 Evaluation

The design of *Move'n'Hold* separates translation and rotation (**Objective 4**), allows always holding the device with two hands (**Objective 5**), supports spatial input (**Objective 6**), and is applicable to both object translation and rotation (**Objective 7**). To investigate how easy and intuitive it is for novel users to learn and apply *Move'n'Hold* and to evaluate its scalability (**Objective 8**) and controllability (**Objective 9**), we conducted two experiments; a main study in which the participants learned the translation prior to the rotation technique and a follow-up study in which the participants learned the rotation prior to the translation technique.

5.3.3.1 Experimental Design

In total, 31 participants (16 male / 15 female, 21-36 years old) were recruited for the evaluation of *Move'n'Hold* for MR-HHDs. All participants had previous experience with HHDs like smartphones or tablets, but only 7 participants had experience with MR and only 6 had experience with MR-HHDs. The experiments involved different task configurations to evaluate *Move'n'Hold*'s effectiveness and flexibility for different complexity levels, directions, and distances. In total, each participant completed 50 translation tasks and 50 rotation tasks.

The participants answered the QUESI (Questionnaire for the subjective consequences of intuitive use) [126] and NASA TLX (NASA task load index) [125] for translation and rotation tasks performed in the experiment. Learnability was assessed based on task completion times (TCTs). In addition, the participants rated their agreement with statements regarding the translation/rotation technique's learnability, suitability for the tasks, conformity with their expectations, satisfaction, and future use. Eventually, the participants provided feedback on the usefulness of manipulation with Hold, as well as on their walking and axis locking preferences.

In a pre-study it turned out that most participants preferred learning translation prior to rotation. Therefore, we let the 20 participants in the main study perform the tasks in this order. To investigate the effects of this order, we conducted a follow-up study with 11 new participants who performed rotation tasks prior to translation tasks.

5.3.3.2 Tasks

The MR scene for translation tasks included a blue and an orange pair of boxes: An opaque manipulable box (15cm × 15cm × 15cm) and the corresponding transparent target box (25cm × 25cm × 25cm). In the experiment, the manipulable boxes had to be moved into the corresponding target boxes. As soon as this was the case, the next task block was started and two new manipulable boxes appeared at new positions. The scene for rotation tasks was set up similarly with two pairs of manipulable boxes (10cm × 15cm × 10cm) and target boxes (20cm × 30cm × 20cm). The manipulable boxes with differently colored sides were placed inside their transparent target boxes and had a different starting orientation. During the experiment, the participants had to rotate the manipulable boxes such that their orientation and the colored sides aligned with the corresponding target

box. After both pairs of boxes were solved, two new manipulable boxes with different orientations appeared automatically.

Both studies were based on four task configuration parts (A, B, C, D) which differed in terms of distance, direction, and complexity as described in Table 5.1. Each part consisted of a number of task blocks. In each task block, the starting position/orientation of the manipulable boxes was computed relative to the target boxes' position and orientation by subtracting (left box) or adding (right box) 30cm (short), 50cm (medium), or 70cm (long) for translation or 20 degree (short), 40 degree (medium), 60 degree (long) for rotation according to the specified complexities and directions. For example, in part A and D, the position/orientation of the manipulable box was only adapted in one dimension. This means that solving a task required manipulations along/around the x-, y-, or z-axis. Respectively, part B and C required manipulations in two and three dimensions. Examples for 1D, 2D, and 3D translation and rotation tasks are given in Fig. 5.6.

Table 5.1: Task configurations.

part	**#task blocks**	**complexity**	**direction**	**distance**
A	6	1D	x-, y-, or z-axis	short (task blocks 1-3), long (task blocks 4-6)
B	3	2D	xy-, xz-, or yz-plane	medium
C	1	3D	x-, y-, and z-axis	medium
D	3	1D	x-, y-, or z-axis	short

5.3.3.3 Procedure

At the beginning of an experiment, the participants were introduced to *Move* (i.e., object manipulation through direct mapping using only left-thumb-touch) and performed a short training session. Afterwards, Part A was completed with *Move*. Next, *Move'n'Hold* was introduced and practiced in another training session. Then, Part A, B, and C were completed with *Move'n'Hold*. To explore how the learnability of device-based object manipulation is affected by the order in which translation and rotation techniques are taught, the participants repeated Part A again with *Move* in the end. While completing Parts A, B, and C the participants were allowed to walk around during one task block but were required to come back to the starting position when a new set of tasks appeared. To investigate the effect of different directions of device movement, the participants were asked to stay in the starting position while completing Part D with *Move* in the end. Participants were allowed to skip tasks if they found them too difficult. The procedure

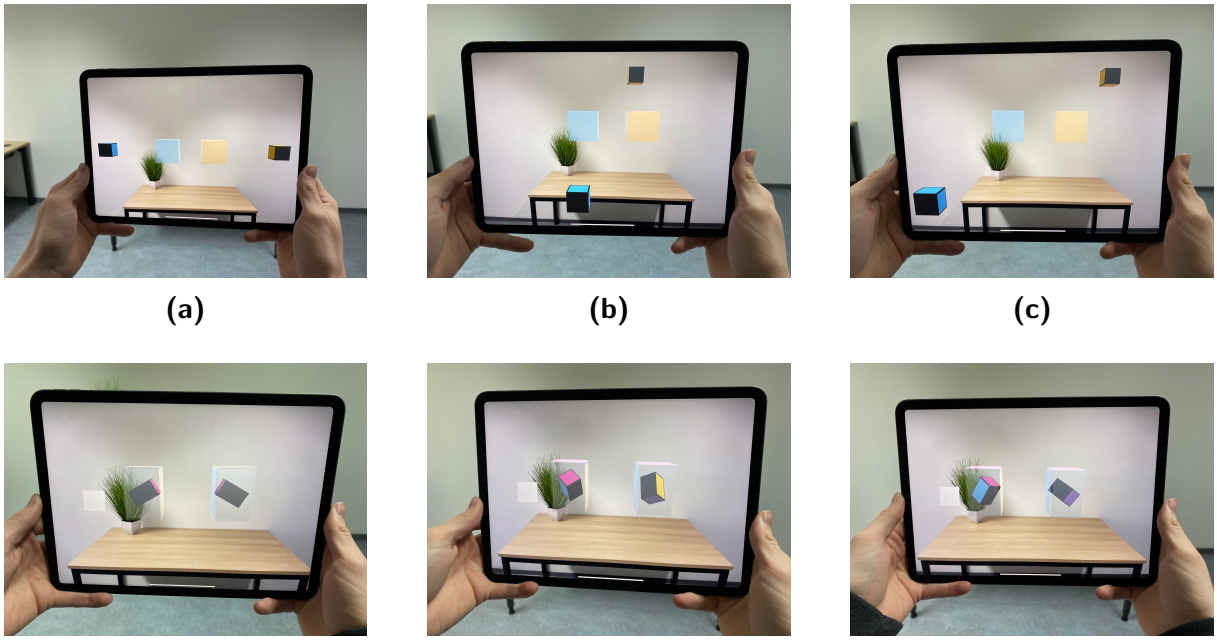


Figure 5.6: Examples for 1D, 2D, and 3D translation (a – c) and rotation (d – f) tasks.

described was followed for both translation and rotation tasks. Throughout the experiments we measured TCTs. After task completion, the participants provided feedback through several questionnaires as described in Chapter 5.3.3.1.

5.3.4 Results

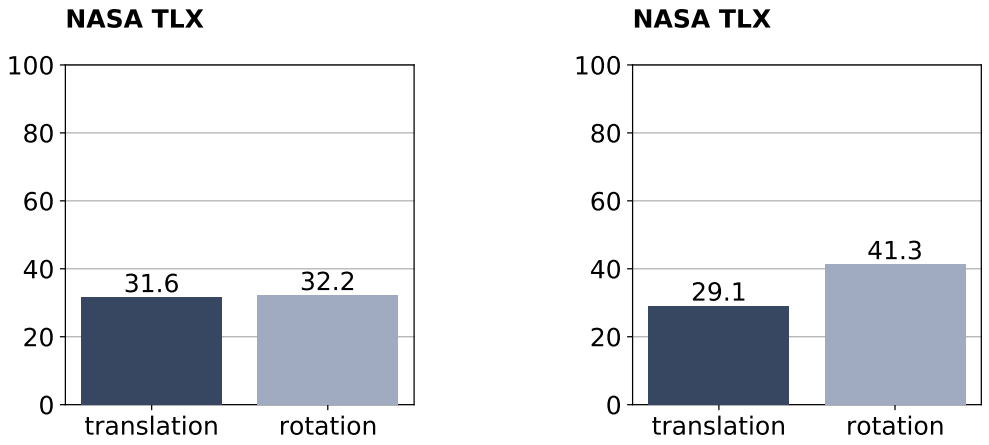
The following sections summarize the findings from the main study and follow-up study.

5.3.4.1 Main Study

All tasks were completed successfully and no participant took up the offer to skip a task because it was perceived too difficult. Therefore, we rate *Move'n'Hold* to be effective for performing object translations and rotations of different complexities, directions, and distances. When comparing the completion times for each task block in Part D, we found that basic manipulations (*Move*) can be performed equally fast in all directions, as a repeated measures anova showed no significant differences between the TCTs for translations ($p > 0.3$) along or rotations ($p > 0.1$) around the x-, y-, and z-axis. Moreover, the mean completion time from the first to the last repetition of Part A decreased significantly by 42% for translation ($p \leq 0.0001$) and 43% for rotation ($0.001 < p \leq 0.01$) in paired

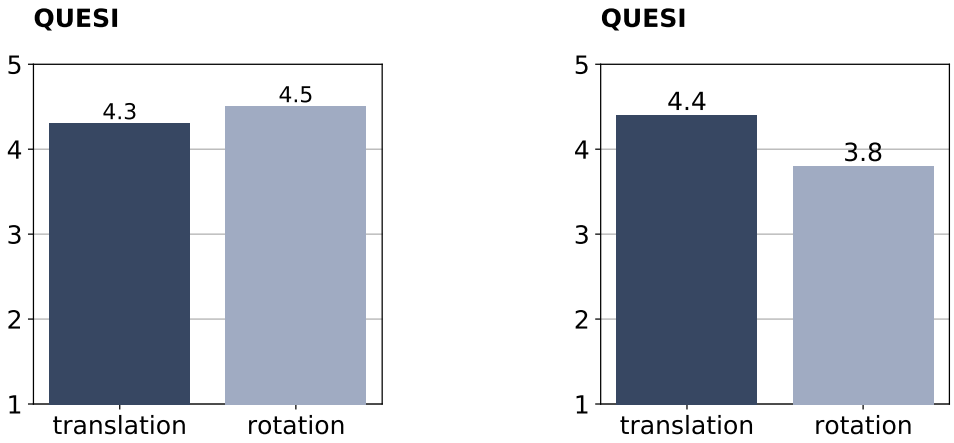
samples t-tests with Bonferroni correction. The standard deviation among the participants' TCTs decreased by 25% for translation and by 64% for rotation tasks, showing that performance discrepancies between users are reduced quickly.

The computed scores for QUESI [126] and NASA TLX [125] reveal high intuitiveness and low workload for translation and rotation. The obtained NASA TLX scores for translation and rotation in the main study (i.e., 31.6 for translation and 32.2 for rotation, Fig. 5.7a) are lower than in at least 80% of the studies reviewed by Grier [70]. The obtained QUESI scores (see Fig. 5.8a) indicate that the participants in the main study perceived both translation (score: 4.3) and rotation (score: 4.5) highly intuitive.



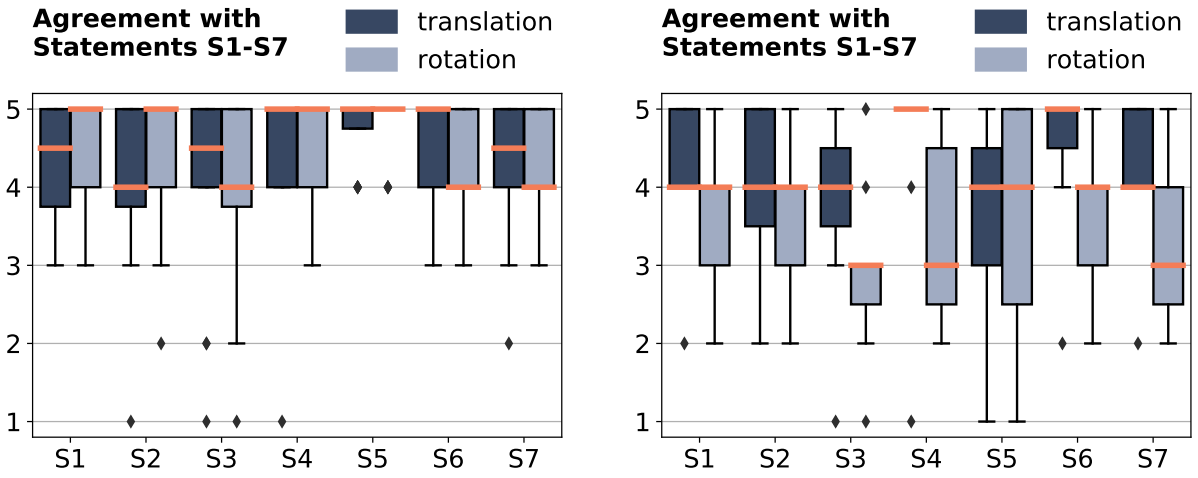
(a) main-study **(b)** follow-up study

Figure 5.7: Mean weighted NASA TLX scores ([0 , 100] with 0 = best, 100 = worst) obtained for the performed object translation and rotation tasks.



(a) main-study **(b)** follow-up study

Figure 5.8: QUESI scores ([1, 5] with 1 = worst, 5 = best) obtained for the performed object translation and rotation tasks.



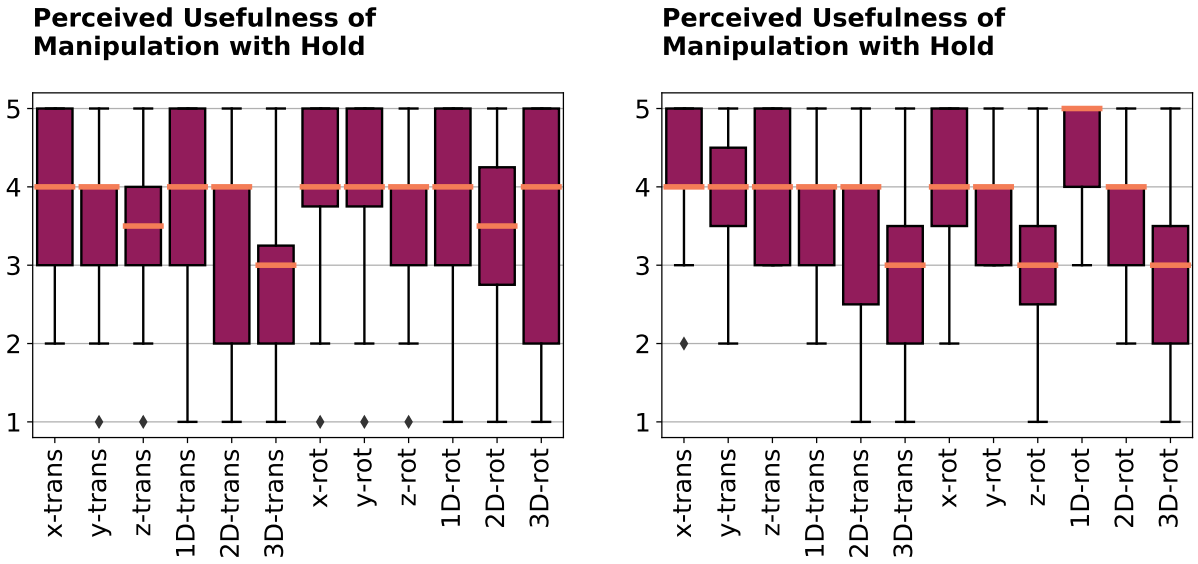
(a) main study **(b)** follow-up study

Figure 5.9: Participants' agreement with statements S1 – S7 from 1 (fully disagree) to 5 (fully agree) regarding the performed object manipulation tasks: **S1** The translation/rotation technique is well suited to the requirements of the task; **S2** The number of steps to translate/rotate objects is adequate; **S3** When translating/rotating objects, I have the feeling that their behavior is predictable; **S4** Learning the translation/rotation technique was very easy; **S5** Relearning the translation/rotation technique after a lengthy interruption will be easy; **S6** Overall, I am satisfied with this translation/rotation technique; **S7** If I had to translate/rotate virtual objects in the future, I would like to use this technique.

The participants provided further feedback by rating their agreement with **S1-S7** (see caption of Fig. 5.9a). The results show that the participants perceived the interaction technique to be easy to learn (**S4**) and thought it will be easy to relearn after a lengthy interruption (**S5**). While using the interaction technique to manipulate objects, they had the feeling that their behavior is predictable (**S3**). The interaction techniques were rated suitable for the tasks (**S1**) and the number of steps to complete a task was rated adequate (**S2**). Overall, the participants were satisfied with the interaction techniques (**S6**) and would like to use them for translating or rotating objects in the future (**S7**).

The participants of our main study found automated continuous manipulation such as available in *Move'n'Hold* most useful for long-distance translations and rotations. Furthermore, Fig. 5.10a shows that manipulation with Hold was rated slightly more useful for 1D tasks compared to 2D and 3D tasks. It was rated most useful for translations along the x-axis and rotations around the x-axis and y-axis.

During the experiments, we observed that the participants approached the tasks in different manners. While some walked around a lot, others mostly stayed at the starting



(a) main study **(b)** follow-up study

Figure 5.10: Perceived usefulness of manipulation with Hold rated from 1 (not useful at all) to 5 (very useful) for performing translations/rotations around the x-, y-, and z-axis as well as for performing 1D, 2D, and 3D translations/rotations.

position. This observation is also reflected in the answers regarding the walking preferences. While translating objects, 60% of the participants preferred to walk around, 30% preferred to not walk around, and 10% did not have a preference. During rotation, 50% preferred walking, 45% preferred to not walk around, and 5% did not have a preference.

Regarding axis locking during rotation, the majority of the participants (85%) preferred to not lock axes and rotate objects around multiple axes at once. Regarding axis locking for translation, the majority of participants stated that if such a feature would be available, they would not use it (50%) or use it less often (20%) than for rotation.

Another difference in the interaction style that we observed concerns the device movements performed to manipulate the objects. Some participants translated or rotated the objects along or around single axes one after the other. On the contrary, other participants performed diagonal translations and rotated objects around multiple axes at once. For object rotation, a particularly smart approach was used by some participants who aligned the HHD with the manipulable object's front side and then rotated the HHD such that it aligns with the target object's front side. In this way, the manipulable box can be intuitively rotated towards the correct target orientation without having to think a lot about the axes around which the objects need to be rotated. Furthermore, we observed that during rotations around the y-axis, some participants actually walked around the ob-

ject to perform the rotation. Regarding the option for automated continuous movements, we observed that Hold was often used to move distant objects closer to the HHD and to correct unintended actions during task completion.

5.3.4.2 Follow-up Study

To investigate whether learning device-based translation before rotation is indeed easier, we conducted a follow-up study in which a new set of participants learned rotation first.

Fig. 5.7 shows that the NASA TLX for translation was slightly lower in the follow-up study (29.1) compared to the main study (31.6). However, for rotation the NASA TLX was clearly higher when performed prior to translation (41.3) than if the translation technique had been learned before (32.2). The obtained QUESI scores show that the second manipulation mode was rated more intuitive in both studies. However, the discrepancies between the scores for translation and rotation were higher when rotation was performed first (see Fig. 5.8).

This tendency is also reflected in the agreement with (**S4**) and (**S5**) (see Fig. 5.9). While in the follow-up study learning translation (**S4**) was perceived only slightly easier than in the main study, learning rotation (**S4**) was perceived much easier if translation had been learned before than if rotation had to be learned before having used the translation technique. In the main study, both translation and rotation were perceived to be much easier to relearn (**S5**) than in the follow-up study. As shown in Fig. 5.9 the discrepancies between the agreement with **S1-S7** for translation and rotation were in general higher in the follow-up than in the main study.

Similar to the main study (see Fig. 5.10a) the participants of the follow-up study (see Fig. 5.10b) perceived manipulation with Hold most useful for 1D tasks, translations along the x-axis, and rotations around the x- and y-axis.

Furthermore, we found decreasing learning effects in the second condition for both groups (i.e., learning effects for translation were less strong in the follow-up (24%, $0.0001 < p \leq 0.001$) than in the main study (42%, $p \leq 0.0001$), and learning effects for rotation were less strong in the main (43%, $0.001 < p \leq 0.01$) than in the follow-up study (53%, $0.001 < p \leq 0.01$)).

Based on these findings, we conclude that *Move'n'Hold* provides learnability across different manipulation techniques and recommend teaching translation prior to rotation.

5.4 Extending Move'n'Hold for Head-mounted Displays

Addressing **Research Question 2**, the design of *Move'n'Hold* is refined and extended for MR-HMDs and VR-HMDs. Thereby, two adjustments were made to the original design of *Move'n'Hold* for MR-HHDs: A red selection point was added to the center of the MR-HHD to support object selection (see Fig. 5.11) and axis locking was omitted as it turned out to be less helpful than expected. Then, interaction with *Move'n'Hold* for MR-HHDs, MR-HMDs, and VR-HMDs is compared against a set of state-of-the-art methods.

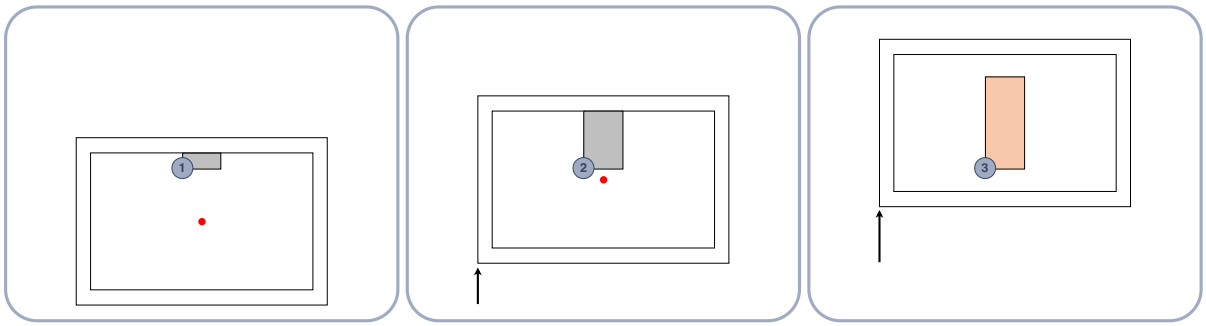


Figure 5.11: The original selection paradigm presented in Fig. 5.2 was extended with a red selection point which serves as a crosshair when selecting objects.

5.4.1 Design

Move'n'Hold involves vision (i.e., centering the desired object on the device's screen) and touch (i.e., moving the tablet while applying peripheral touch). To transfer *Move'n'Hold* as seamlessly as possible to HMDs, we add the same red selection point to the center of the user's view in the HMD and combine the HMD with a tablet controller which implements the same interaction paradigm as described in Chapter 5.3.1.

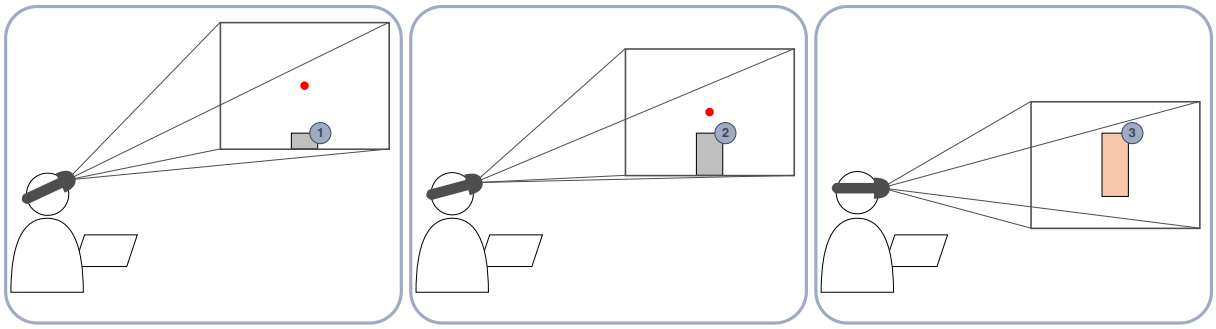
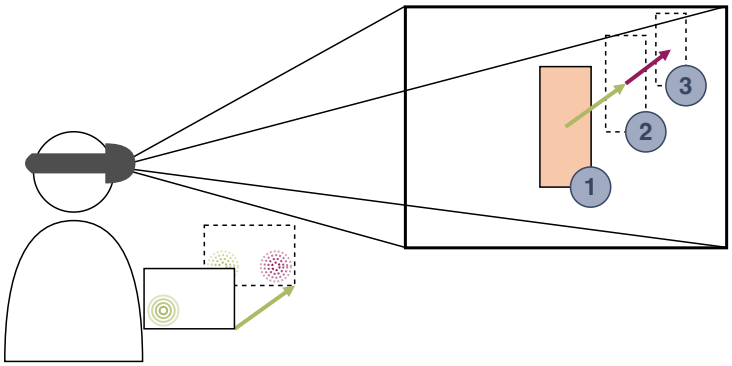
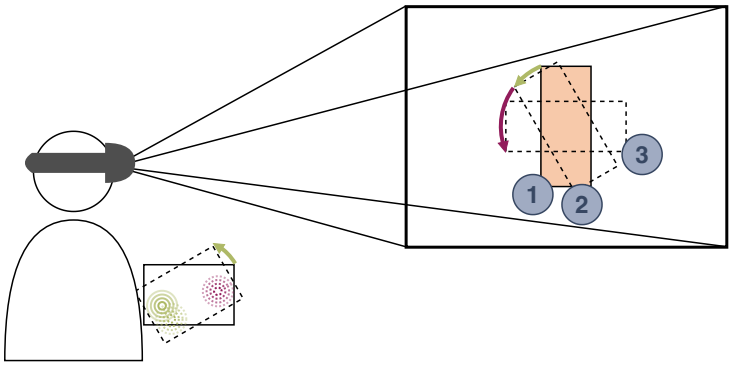


Figure 5.12: To select an object while using *Move'n'Hold* with an HMD, the user has to adapt the HMD's position or orientation such that the object appears at the center of the display and is hit by the red selection point.



(a) Object Translation



(b) Object Rotation

Figure 5.13: To (a) translate or (b) rotate an object seen through the HMD the user has to apply touch on the tablet controller while moving the device. As long as only left-thumb-touch is active the tablet's movement in space is directly mapped to the object. By adding right thumb touch, automated object movment is started.

As such, an object can be specified through vision (i.e., moving the red selection point by adjusting the HMD's field of view, see Fig. 5.12) and manipulated through the combination of touch input and tablet movement as illustrated in Fig. 5.13. When using *Move'n'Hold* for MR-HHDs, the tablet handles both input and output and thus has to be held up high. On the contrary, in *Move'n'Hold* for HMDs, the tablet only handles input and therefore can be held in any position and orientation.

5.4.2 Implementation

To evaluate the set of consistent object manipulation techniques offered by *Move'n'Hold* we compared it to a set of state-of-the-art (*SotA*) interaction methods. To this end, we implemented a total of 12 interaction techniques (i.e., system {*Move'n'Hold*, *SotA*} × manipulation mode {translation, rotation} × device {MR-HHD, MR-HMD, VR-HMD}).

The applications were developed in Unity using ARKit, Mixed Reality Toolkit, and XR Interaction Toolkit. The MR-HHD apps were deployed to an 11-inch Apple iPad Pro Gen. 3, the MR-HMD apps were deployed to Microsoft HoloLens 2, and the VR-HMD apps were displayed on an HTC VIVE Pro. The tablet controller for using *Move'n'Hold* on the MR-HMD and VR-HMD was an 11-inch Apple iPad Pro Gen. 4. An overview of the interaction techniques and settings offered by both systems is given in Table 5.2 and Fig. 5.14.

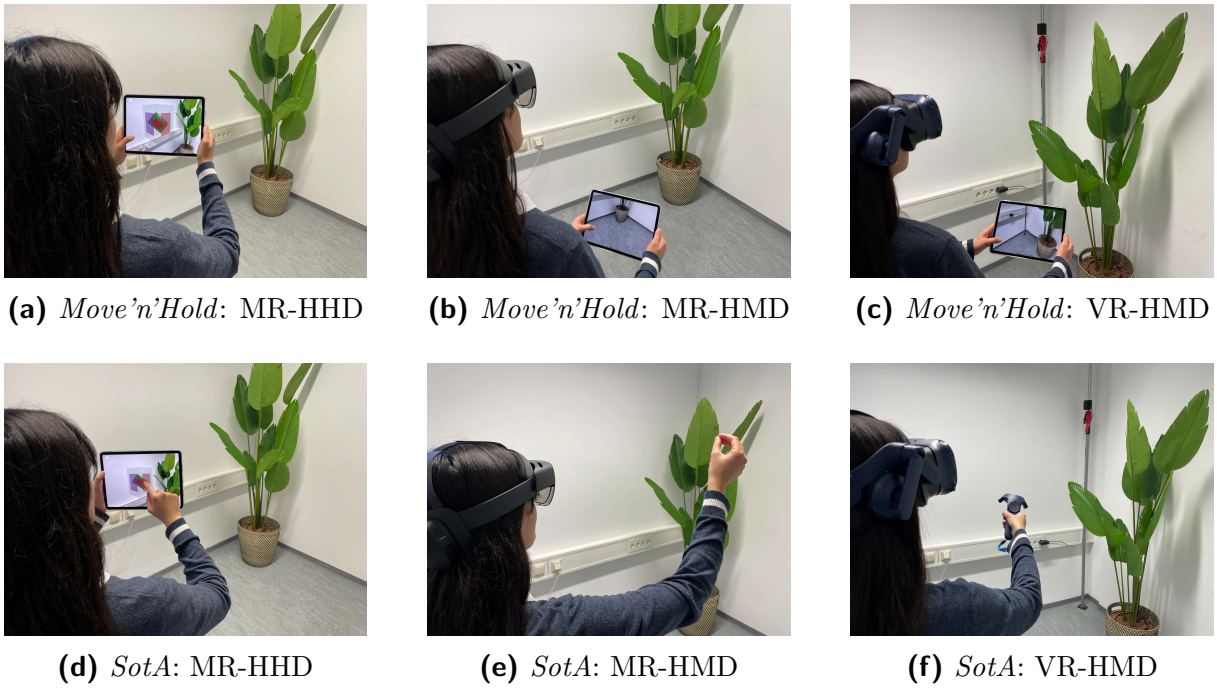
Table 5.2: Methods for object selection, translation, and rotation as provided by *Move'n'Hold* and *SotA*.

Move'n'Hold *MR-HHD, MR-HMD, VR-HMD*	***SotA*** *MR-HHD*	*MR-HMD*	*VR-HMD*
Selection			
adjust the position or orientation of the HHD / HMD such that the object is hit by the red selection point at center of the display	touch the object on the HHD's screen	point at the object with the finger (hand gesture)	point at the object with the controller
Translation / Rotation			
while left-thumb-touch is applied, the tablet's translation / rotation is mapped to the object; by adding right-thumb-touch, continuous translation / rotation can be started	the object is translated / rotated by dragging the finger on the HHD's screen	the object is translated / rotated based on hand movements while a pinch gesture is performed	the object is translated / rotated based on controller movements while the trigger button is pressed

5.4.2.1 Move'n'Hold Interaction Methods

Move'n'Hold for the MR-HHD was implemented as described in Chapter 5.3.2 with two exceptions: A red selection point was added as a crosshair to aid object selection and axis locking was not implemented as it turned out to be less helpful than expected.

The apps for *Move'n'Hold* with the MR-HMD and the VR-HMD consist of two sub-applications each which communicate through UnityWebRequests: the tablet application which handles user input and the HMD application which handles the output. To align the tablet's and the HMD's coordinate systems, both devices are started in a fixed position



(a) *Move'n'Hold*: MR-HHD **(b)** *Move'n'Hold*: MR-HMD **(c)** *Move'n'Hold*: VR-HMD

(d) *SotA*: MR-HHD **(e)** *SotA*: MR-HMD **(f)** *SotA*: VR-HMD

Figure 5.14: Overview of the interaction settings for object manipulation using *Move'n'Hold* and *SotA*.

and orientation. Upon launch the tablet application asks the HMD application to share the active manipulation mode which is either translation or rotation.

While the tablet registers no touch, the HMD application checks for collisions between a ray originating from the HMD and manipulable objects. The red selection point which is always located at the display's center thereby serves as a crosshair. Upon the detection of a collision, the selection point disappears and the object turns green. When touch is detected on the tablet, the tablet application asks the HMD for currently selected objects.

If an object was hit by the ray originating from the HMD when left-thumb-touch was registered, the tablet starts to cache its movement: v_{move}, q_{move}, and *hold* (a boolean which stores if right-thumb-touch is registered). If *hold* is true, v_{hold} / q_{hold} are calculated and cached as well (Eq. 5.3 / Eq. 5.4). The data collected on the tablet is sent to the HMD in every frame. The HMD then constantly updates internal variables storing *hold*, v_{hold} / q_{hold}, and adds v_{move} / q_{move} to two lists. While only left-thumb-touch is active, the objects are manipulated by directly mapping the tablet's movements to the objects. In translation mode, the HMD application therefore processes the list of vectors v_{move} and adds them to the current object position v_{obj} (Eq. 5.1). Analogously in rotation mode, the HMD application processes the list of quaternions q_{move} and multiplies them by the current object orientation q_{obj} (Eq. 5.2). When touch is registered on both display sides,

the object's new position v_{lerp} / orientation q_{lerp} is computed based on v_{hold} / q_{hold} and linear interpolations (Eq. 5.5 / Eq. 5.6). As soon as touch is released on both sides, the tablet informs the HMD accordingly, selection with the ray originating from the HMD is activated again, object manipulation stops, and the lists of vectors / quaternions storing the tablet's movements in the HMD are cleared.

As such, the same interaction paradigm is provided for MR-HHDs, MR-HMDs, and VR-HMDs (see Fig. 5.15).

5.4.2.2 State-of-the-Art Interaction Methods

For the *SotA* methods we chose to implement out-of-the-box techniques based on touch, mid-air gestures, and controllers which are most common in practical use. Similar to the *Move'n'Hold* methods, we separate translation and rotation.

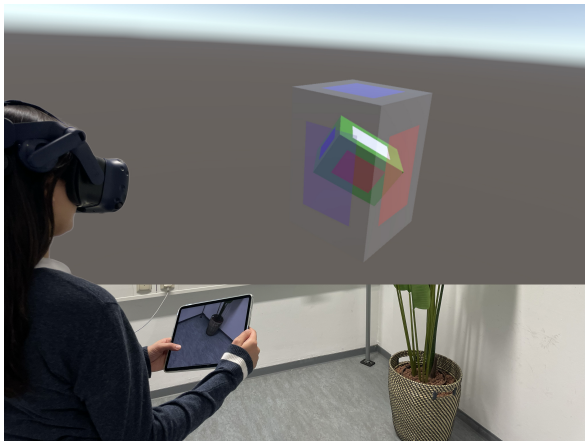
Using the MR-HHD application, objects can be selected via touch. Upon selection, they turn green and can be manipulated by dragging the finger on the screen. For translation tasks, we track the movement of the finger touching the screen. To this end, we use an invisible plane which is parallel to the tablet's orientation. The selected object is translated according to the detected movements of the finger on the invisible plane. For touch-based rotation, we use an invisible sphere which surrounds the selected object. The user can move the finger on the touchscreen to rotate the invisible sphere around its center. The sphere's rotation is then applied to the object.

For the MR-HMD we enable object manipulation with the standard mid-air gestures for Microsoft HoloLens 2 from the Mixed Reality Toolkit. To manipulate an object, users can point at objects with the hand. As soon as the ray originating from the hand collides with the object, the user has to perform a pinch gesture and then move the hand. While the pinch is performed, the object will follow the hand's translation or rotation in space depending on the active manipulation mode.

For the VR-HMD we provide object manipulation with the standard HTC VIVE Pro controller. To this end, we extract controller input, i.e., a user pressing the trigger button, the controller's position and orientation. The user can select an object by pressing the controller's trigger button when a ray originating from the controller collides with the object. Depending on the active manipulation mode, the controller's translation or rotation is then mapped to the object as long as the trigger button is pressed.

(a) Example MR-HHD: Continuous object translation using left-thumb-touch and right-thumb-touch.

(b) Example MR-HMD: Selecting an object by hovering it with the red point while no touch is applied on the tablet controller.



(c) Example VR-HMD: Object rotation though direct mapping using left-thumb-touch on the tablet controller.

Figure 5.15: *Move'n'Hold* provides the same object manipulation paradigm for MR-HHDs, MR-HMDs, and VR-HMDs. Each example action shown in (a), (b), and (c) for one technology is likewise available for the other two.

5.4.3 Evaluation

The extended version of *Move'n'Hold* provides the same set of operations for all access points of XR^S (**Objective 1**) and allows performing the same operation (i.e., object translation or rotation) with the same interaction paradigm on all devices (**Objective 2**). To investigate the existence of cross-device benefits (**Objective 3**) and compare *Move'n'Hold* to the set of state-of-the-art interaction techniques a detailed user study was conducted.

5.4.3.1 Experimental Design and Procedure

For the comparative study, 20 participants (10 male / 10 female, 20-37 years old) were recruited. Some of the participants had used MR-HHDs (70%), MR-HMDs (40%), and VR-HMDs (60%) prior to our study. Three independent variables were considered in the experiment: system {*Move'n'Hold*, *SotA*}, device {MR-HHD, MR-HMD, VR-HMD}, and task {translation, rotation}. In total, the experiment therefore consisted of 12 sessions (system × device × task); one session for each of the developed interaction techniques.

At the beginning of each session, the participants watched an explanatory video of the respective interaction technique and performed a short training session. The order in which the sessions were conducted was as follows. Half of the participants performed all tasks with *Move'n'Hold* prior to *SotA* and vice versa. For both systems, translation and rotation tasks were first completed with the MR-HHD. In order to investigate cross-device learnability (i.e., how the usage of one device benefits the use of another device), half of the participants continued with the VR-HMD and completed the tasks with the MR-HMD last while the other half used the HMDs in reverse order. Taking into account the results from the previous study, translation was always performed prior to rotation.

Several dependent variables were considered to compare *SotA* and *Move'n'Hold*. After each session, the participants provided two difficulty ratings for the respective interaction technique. The first difficulty rating (DIFF_exp) refers to the perceived difficulty while performing the tasks in our experiments and the second rating (DIFF_large) refers to the expected difficulty while performing larger manipulations (i.e., longer translations/rotations). After all interaction techniques for the first system had been used (i.e., six sessions), the participants answered questionnaires to assess the system with the System Usability Scale (SUS) [21] and the NASA TLX [125]. The same procedure was followed for the second system. After task completion with both systems, we asked the participants about the system which they (Q1) prefer, (Q2) think provides higher accuracy, (Q3) more cross-device benefits, and (Q4) expect to be easier to relearn. Throughout the experiment, we measured task completion times (TCTs).

5.4.3.2 Tasks

In each of the 12 sessions, 8 translation or rotation tasks were performed with one of the three devices using one system. Thus, every participant completed 96 manipulation tasks.

Translation tasks required the participant to move an opaque manipulable cube (0.2m × 0.2m × 0.2m) into a semi-transparent target cube (0.25m × 0.25m × 0.25m) which was placed in the center of the scene. As soon as the manipulable cube was inside the target cube, the manipulable cube disappeared and the next manipulable cube appeared at a new position. The initial positions of the manipulable cubes were computed based on the target cube's position, which was always the same. Starting from the position of the target cube, the manipulable cube was translated 0.75m along all axes (x, y, and z) either in the positive or negative direction, yielding 8 different positions.

The rotation tasks were designed similarly. An opaque box (0.1m × 0.15m × 0.1m) was placed inside a semi-transparent target box (0.2m × 0.3m × 0.2m) in the scene's center. Both boxes had differently colored sides. To complete a task, the inner box had to be rotated such that its orientation (and the colored sides) aligned with the one of the target box. As soon as the inner box's orientation differed less than 4 degrees on each axis from the orientation of the target box, the task was solved and the next box appeared. Analogously to the translation tasks, the initial orientations of the manipulable boxes were computed based on the target box's orientation. The orientation of the manipulable boxes were set by 40 degree rotations around all axes either in the positive or in the negative direction, yielding again 8 different orientations.

Before the first task started, a simple task (i.e., either a 0.75m translation along or a 40 degree rotation around the x-axis) had to be completed to ensure the comparability of task completion times (TCTs). During the experiment, the participant was allowed to move around in a limited area (3m × 3m).

5.4.3.3 Hypotheses

The following hypotheses were formulated to compare *Move'n'Hold* to *SotA*.

- **H1** Overall, translation and rotation tasks can be completed faster with *Move'n'Hold* than with *SotA*.
- **H2** Participants who use the MR-HMD after the VR-HMD, complete (a) translation and (b) rotation tasks with the MR-HMD faster than participants who use the MR-HMD first. Participants who use the VR-HMD after the MR-HMD, complete (c) translation and (d) rotation tasks with the VR-HMD faster than participants who

use the VR-HMD first. These improvements are higher for *Move'n'Hold* compared to *SotA*.

- **H3** The perceived difficulty of performing object (a) translations and (b) rotations in our study is lower with *Move'n'Hold* than with *SotA*. The expected difficulty of performing longer (c) translations and (d) rotations with *Move'n'Hold* is lower than with *SotA*.

- **H4** The NASA TLX scores for (a) translation and (b) rotation tasks are lower for *Move'n'Hold* than for *SotA*.

- **H5** The SUS scores for *Move'n'Hold* are higher than for *SotA*.

- **H6** If their job would require them to use and switch between MR-HHDs, MR-HMDs, and VR-HMDs, the participants will prefer to use *Move'n'Hold* instead of *SotA*.

- **H7** The participants think relearning *Move'n'Hold* will be easier than relearning *SotA*.

- **H8** If they had to complete similar tasks as accurately as possible, the participants think they could achieve the best results using *Move'n'Hold* instead of *SotA*.

- **H9** The participants think that *Move'n'Hold* provides more cross-device benefits than *SotA*.

In the following, we refer to the numbering of these hypotheses and highlight them according to the level of support they received in **green** (high support), **orange** (medium support), and **red** (no support).

5.4.4 Results

To compare the temporal effort when using *Move'n'Hold* and *SotA*, we measured the time between the appearance and the disappearance of each virtual box as TCT. However, the mean sum of TCTs for completing all (i.e., translation and rotation tasks) with all devices was only slightly lowever with *Move'n'Hold* than with *SotA* (see Fig. 5.16, **H1**).

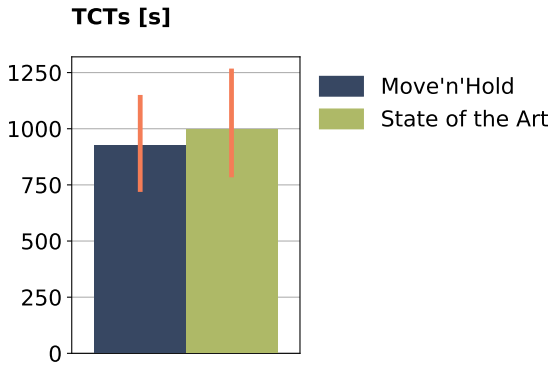


Figure 5.16: Mean total TCTs [s] of all translation + rotation tasks and 95% confidence intervals for task completion with *Move'n'Hold* and *SotA*.

To examine cross-device learnability (i.e., the extent to which users benefit from prior use of one device when completing tasks with another) half of the participants (GroupA) used the MR-HMD after the VR-HMD and the other half (GroupB) used the VR-HMD after the MR-HMD (see Fig. 5.17).

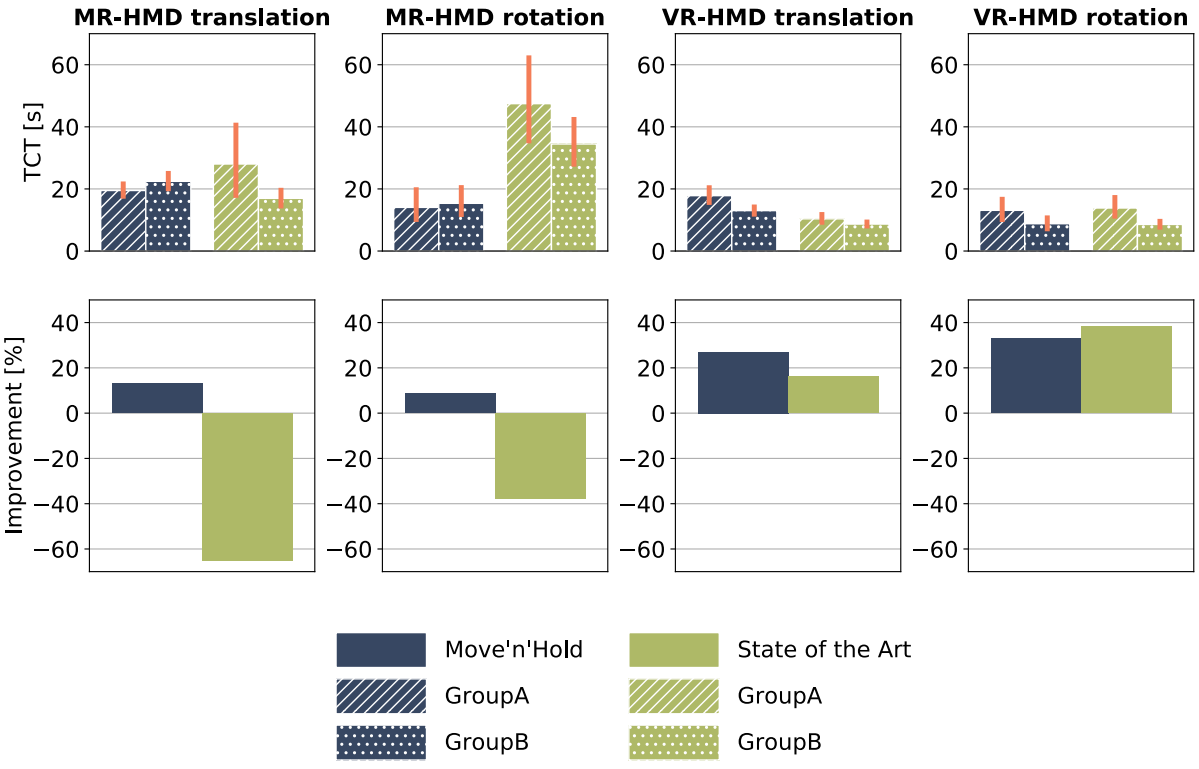


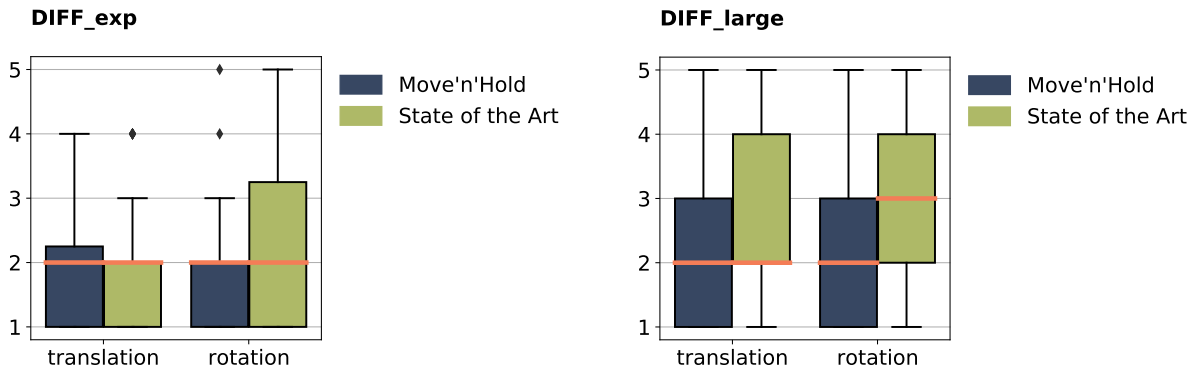
Figure 5.17: Cross-Device Learnability Results: The first row shows mean TCTs and 95% confidence intervals for single translation / rotation tasks completed with the MR-HMD / VR-HMD by GroupA (VR-HMD before MR-HMD) / GroupB (MR-HMD before VR-HMD); the second row shows the improvement of the mean TCTs from (MR-HMD) GroupB to GroupA and (VR-HMD) GroupA to GroupB.

When comparing the mean TCTs of these groups for single translation and rotation tasks (see Fig. 5.17), we found that *Move'n'Hold* provided higher overall improvements than *SotA* when moving from one HMD to the other.

Fig. 5.17 further shows that GroupA was slower than GroupB when using the MR-HMD with *SotA*, which results in a negative value in the chart showing the cross-device improvement. When using *Move'n'Hold* with the MR-HMD, GroupA had lower TCTs when translating (first column in Fig. 5.17, **H2a**) and rotating (second column in Fig. 5.17, **H2b**) objects than GroupB. When using the VR-HMD, GroupB was faster than GroupA using *SotA* and *Move'n'Hold*. Thereby, higher improvements were observed for *Move'n'Hold* during translation (third column in Fig. 5.17, **H2c**) and for *SotA* during rotation (fourth column in Fig. 5.17, **H2d**).

Regarding the difficulty ratings for the tasks completed in the experiments, Fig. 5.18a shows that *Move'n'Hold* was perceived slightly more difficult than *SotA* for translation (**H3a**). However, *SotA* was rated substantially more difficult than *Move'n'Hold* for rotation (**H3b**).

Moreover, our participants expected *Move'n'Hold* to be less difficult than *SotA* during large translations and rotations (see Fig. 5.18b, **H3c+d**).



(a) Object translations and rotations performed in the experiment.

(b) Larger object translations and rotations than in the experiment.

Figure 5.18: Rated difficulty ([1, 5] with 1 = very easy, 5 = very difficult) for using *Move'n'Hold* and *SotA*.

The NASA TLX scores for both systems show that *Move'n'Hold* outperformed *SotA* in the translation and rotation tasks (see Fig. 5.19, **H4a+b**). Similar to the difficulty ratings, *Move'n'Hold* substantially decreased the workload experienced during rotation compared to *SotA*.

Considering Grier's [70] meta-analysis, the total weighted NASA TLX score obtained for *Move'n'Hold* (20.88) is lower than 90% of the UIs reviewed while for *SotA* (33.73) it is only lower than 75% of the UIs reviewed.

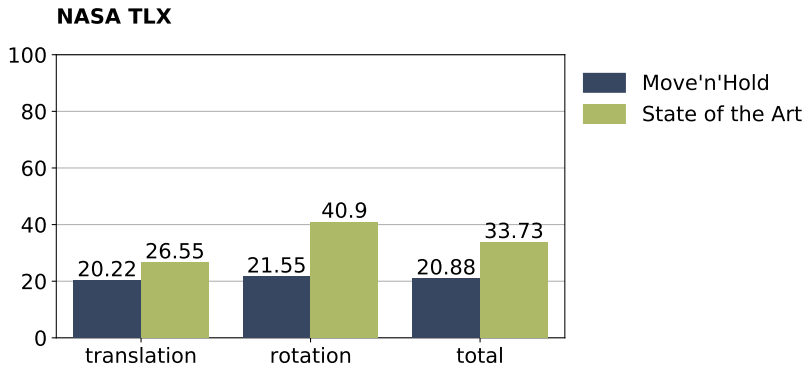


Figure 5.19: Mean weighted NASA TLX scores ([0, 100] with 0 = best, 100 = worst) obtained for object manipulation using *Move'n'Hold* and *SotA*.

On top of that, the computed SUS scores for *Move'n'Hold* (80.88) and *SotA* (66.13) indicate that *Move'n'Hold* provides higher usability (see Fig. 5.20, **H5**). Considering the adjective ratings for SUS in [6], *Move'n'Hold*'s SUS score indicates *Good* to *Excellent* user-friendliness while *SotA*'s SUS score only indicates *OK* to *Good* user-friendliness.

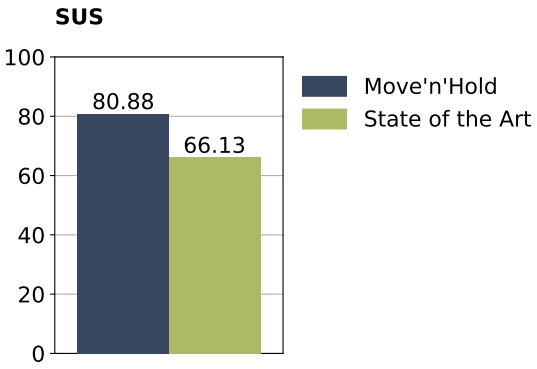


Figure 5.20: Mean SUS scores ([0, 100] with 0 = worst, 100 = best) obtained for object manipulation with *Move'n'Hold* and *SotA*.

As shown in Fig. 5.21, the majority chose *Move'n'Hold* as their preferred system (**H6**), thinks that it will be easier to relearn (**H7**), provides more accuracy (**H8**) and cross-device benefits (i.e., benefits gained from using one device when switching to another) (**H9**).

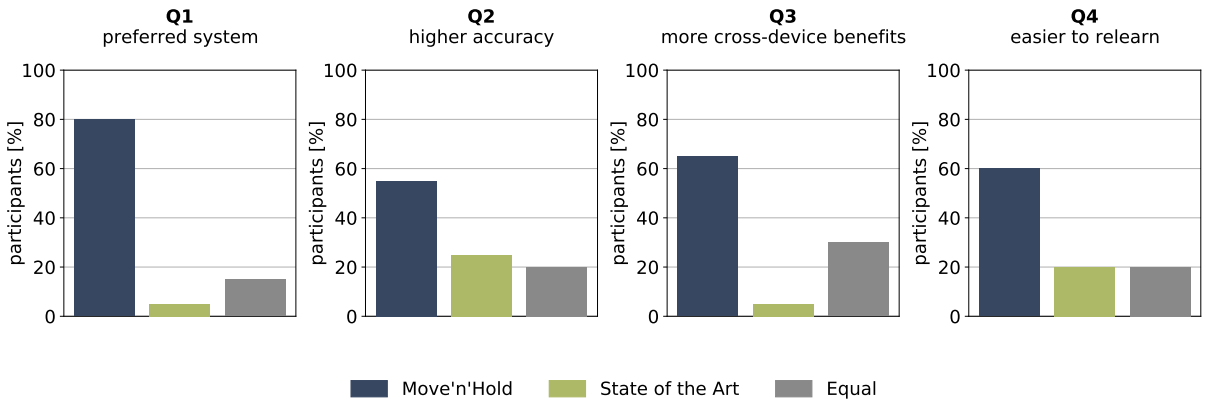


Figure 5.21: Participants' responses [%] to the final comparison of *Move'n'Hold* and *SotA*.

5.5 Discussion

To make it easier for users of XR^S spaces to interact with virtual objects through all of its access points (i.e., MR-HHDs, MR-HMDs, and VR-HMDs) and to reduce overheads when switching between technologies, this chapter introduces the design of a novel object manipulation paradigm which allows performing object translations and rotations with the same interaction paradigm for all access points of XR^S (**Objective 1+2**).

Starting with MR-HHDs, Chapter 5.3 presents the highly scalable novel object manipulation technique *Move'n'Hold* which allows translating and rotating objects through seamless combinations of natural manipulation through direct mapping of device movements when only left-thumb-touch is applied with automated continuous manipulations which are started or stopped when right-thumb-touch is added or released. As such, *Move'n'Hold* supports small and large, fine and coarse, slow and fast manipulations. At the same time, the user retains full control over the activation and speed of continuous movement. In Chapter 5.4, the design of *Move'n'Hold* for MR-HHDs was refined and transferred to MR-HMDs and VR-HMDs which implement the same interaction paradigm using a tablet controller.

With *Move'n'Hold* an object can be translated and rotated separately (**Objective 4**) with the same interaction paradigm (**Objective 7**). The combination of peripheral touch input and device movements allows spatial input (**Objective 6**) while holding the MR-HHD or the tablet controller (when using MR-HMDs or VR-HMDs) with both hands at all times (**Objective 5**).

The results of the first evaluation in Chapter 5.3, did not only reveal *Move'n'Hold* to be easy and intuitive to learn but also demonstrated that the knowledge gained while translating objects helps users to learn and apply the rotation method and vice versa. Thereby, the overall experience was better when the translation technique was learned prior to rotation. Hence, we recommend teaching the manipulation modes in this order and followed this strategy when evaluating the extended version of *Move'n'Hold* against *SotA* in Chapter 5.4. The effective completion of all tasks varying with respect to distance, direction, and complexity further demonstrate *Move'n'Hold*'s scalability (**Objective 8**). On top of that, we observed that *Move'n'Hold* provided high controllability (**Objective 9**) as the participants activated continuous manipulation, adjusted its speed and combined it with natural manipulation through direct mapping according to their individual preferences and interaction styles. These insights were also confirmed by the observations made when evaluating *Move'n'Hold* for HMDs in Chapter 5.4.

Participants of the study in Chapter 5.3 rated manipulation with Hold most useful for 1D tasks. This could indicate that it is easier for a user to predict the position and orientation of an object during continuous manipulation along or around one axis. The perceived usefulness was particularly high for x- and y-axis rotations. In contrast to z-axis rotations, the HHD's display becomes invisible to the user when it is rotated too far around the x- or y-axis. The option for automated continuous manipulation offered by *Move'n'Hold* allows the user to only perform a small rotation around the x- or y-axis and then continue with automated rotations without having to move the device further. In this way, the object remains visible to the user. As such, *Move'n'Hold* solves a major usability issue of existing object manipulation methods and enhances scalability regarding object rotations in different directions.

The comparison of *Move'n'Hold* and *SotA* in Chapter 5.4 showed that the temporal efforts to complete the tasks were only slightly reduced in *Move'n'Hold*. However, based on the cross-device benefits reported by the participants and reflected in the learnability results we expect these temporal efforts for *Move'n'Hold* to decrease further when *Move'n'Hold* is used more extensively and users switch between technologies more often. Furthermore, *Move'n'Hold* substantially reduced the workload, especially while rotating objects.

The comparative study involved the iterative learning of *Move'n'Hold* on three devices. In total, 6 video tutorials for *Move'n'Hold* were provided for each device {MR-HHD, MR-HMD, VR-HMD} $\times$ task {translation, rotation} combination. In the last sessions of task completion with *Move'n'Hold*, many participants reported that they already know how

to use *Move'n'Hold* and do not need the explanatory video. To maintain comparability of the results, the videos still had to be watched by all participants. Nevertheless, these comments confirm that users are able to seamlessly transfer the interaction paradigm provided by *Move'n'Hold* to other devices and it indeed enables seamless switching between the main access points of XR^S. Together with the subjective feedback, stating that users perceived more cross-device benefits using *Move'n'Hold*, we therefore conclude that using *Move'n'Hold* with one device indeed serves as a training phase when switching to another device (**Objective 3**).

In both studies participants independently discovered and pursued two interesting approaches to solve the task. The first one concerns right-thumb-touch which was not only applied to perform long and continuous manipulations. Instead, the participants also repeatedly applied and released touch on the right side to perform short automated movements. The second approach observed in both studies concerns rotation tasks during which participants aligned the tablet's front side with the manipulable box's front side and then moved the tablet towards the target box's front side while applying touch to perform complex rotations even without looking at the tablet such as for example when using the VR-HMD.

While *Move'n'Hold* does not provide hands-free interaction with HMDs, we do not consider this disadvantageous. Other researchers pointed out opportunities which are provided by the integration of mobile devices and HMDs. For example, they can offer secondary output modalities or input through non-spatial UIs [77, 94, 104, 143, 153, 174, 196, 110] as well as a window towards reality when immersed in VR [61]. The comparative study revealed the limitations of hands-free input modalities provided by state-of-the-art methods. Especially while performing object rotations with mid-air gestures the participants experienced difficulties due to movement restrictions in the wrist. We observed similar issues with the controller, however they were not as severe as with the gestures. *Move'n'Hold* addresses this issue through the integration of automated continuous movements. Using other hands-free input modalities such as gaze or speech for 3D manipulation is even more complex. Thus, for 3D object manipulation, the integration of a tablet controller seems more suitable than the state-of-the-art hands-free input modalities. If perfect hand tracking becomes available in the future, it could also be investigated whether the physical tablet can be replaced by an imaginary tablet. For example, the user could then perform tap gestures instead of left- and right-thumb touch while translating and rotating an imaginary tablet.

The virtual boxes which were manipulated in both studies can be easily replaced with other virtual objects such as virtual furniture, machines, or components of virtual prototypes in domains like the automotive or aerospace industry. An example application in which translations and rotations are combined to manipulate factory components with different devices using *Move'n'Hold* is presented in the next chapter.

Parts of this chapter have been previously published in:

V. M. Memmesheimer, K. J. Klingshirn, B. Ravani, and A. Ebert (2023): Move'n'Hold: Scalable Device-Based Interaction for Mixed Reality Handheld Displays. In Proceedings of the European Conference on Cognitive Ergonomics 2023 (ECCE '23). Article 13, pp. 1-8. ACM, New York, NY, USA. doi: 10.1145/3605655.3605656.

V. M. Memmesheimer, K. J. Klingshirn, C. Herold, B. Ravani, and A. Ebert (2024): Move'n'Hold Pro: Consistent Spatial Interaction Techniques for Object Manipulation with Handheld and Head-mounted Displays in Extended Reality. In Proceedings of the European Conference on Cognitive Ergonomics 2024 (ECCE '24). Article 10, pp. 1-8. ACM, New York, NY, USA. doi: 10.1145/3673805.3673814.

Chapter 6

Practical Applications

The contributions of this dissertation apply to many different domains including construction and manufacturing (e.g., in the aerospace, automotive, chemical, or food industry). For instance, XR^S can be used for facility or product design, production planning, and training the operation or maintenance of machines. In this chapter, some of the concepts and UIs presented in this dissertation are implemented and evaluated for robot control and factory layout planning (FLP), showcasing their practical applicability.

Chapter 6.1 presents a summary of research and aspects which are relevant to the two applications: robot control and FLP. Following this, Chapter 6.2 explores robot control through a MR-HHD-UI. In the context of XR^S this relates to the proposed robotic system for manipulating distant, large, heavy, or hazardous items. By manipulating virtual replicas of physical objects through the MR-HHD-UI a robot arm can be commanded to perform the same operation in the real world. In a detailed study we then evaluate the combination of human and robotic capabilities, by comparing our developed MR-HHD-UI to a Gamepad-UI and a Desktop-UI. Chapter 6.3 continues with the second application and presents how concepts of XR^S can be adopted in different stages of FLP. To this end, we perform a comprehensive analysis on the suitability of XR technologies for various use cases within the FLP process. Based on the analysis, *Move'n'Hold* is implemented in a multi-user MR-HHD application for conceptual FLP and a VR-HMD application for detailed FLP. Then, both applications are evaluated in a pilot study from which we derive design guidelines. Eventually, Chapter 6.4 provides a joint discussion of the results.

6.1 Related Research and Aspects

Despite the potential of XR technologies which has been identified to support industrial use cases, its application in practical settings is still limited. In this context, Emporio et al. [46] note that the effective integration of XR in cyber-physical factories requires the development and evaluation of advanced interaction techniques. Addressing this need, this chapter presents interactive XR UIs for two practical examples: robot control and factory layout planning. As a foundation, this section presents the relevant background information.

6.1.1 XR-supported Robot Control

While robots can increase productivity and prevent errors or accidents by continuous, reliable, and precise task completion in several domains including healthcare, homecare, and space exploration, leveraging their full potential requires human intervention to perform complex task planning, supervision, and maintenance. Thus, the development of UIs which streamline interaction with robots for non-experts and avoid unnecessary complexities is vital. Existing UIs include gamepads and desktop applications. However, these require users to decompose high-level tasks (e.g., picking and placing objects) into detailed instructions which can be interpreted and executed by robotic systems. In this context, the application of MR technologies is deemed promising [164]. For example, the robot's operating environment can be virtually augmented to provide a preview for reviewing the effects of a command prior to its execution. The seamless integration of virtual and physical objects thereby allows the operator to perform the review without having to shift focus.

Previous research which considered the application of MR technologies for human robot interaction considered both HMDs [27, 132, 147, 171] and HHDs [24, 51, 54]. The respective applications allow defining points in space for task and path planning [24, 27, 54], controlling the robot by manipulating virtual replicas of physical objects [51, 132, 147], or visualizing the robot's intended movement through the integration of virtual augmentations [171]. Existing MR-HHD-UIs for robot control use touch input. For example, the MR-HHD application from Chen et al. [29] allows users to define target positions and path trajectories using touch-sliders and drag-and-drop touch gestures. Kapinus et al. [88] allow users to program a robot to perform complex processes by connecting virtual

pucks on a touch-based UI. Thereby, physical objects are overlaid with invisible bounding boxes. In this way, users can select the object by touching it on the HHD's screen. In the approach from Frank et al. [52] virtual replicas of physical objects can be manipulated through tapping, dragging, and rotating fingers on the HHD's screen. Furthermore, Chacko and Kapila [25] allow users to move a smartphone to align a crosshair with pick and place locations which are subsequently marked via button click. However, their approach lacks a comparative evaluation with non-MR-based UIs as well as virtual replicas of physical objects which we consider one of MR's key benefits. The integration of virtual replicas allows detecting misplacements prior to execution and is particularly relevant when dealing with differently sized objects.

6.1.2 XR-supported Factory Layout Planning

Factory layout planning (FLP) is a complex task dealing with the optimal arrangement of a factory's functional units as well as the components within these units. The following two paragraphs provide a brief summary of the relevant background information on FLP based on [178, 187, 22].

FLP encompasses different planning scenarios which can be divided into brownfield and greenfield planning. While greeenfield planning involves planning a completely new factory, brownfield planning is focused on restructuring an existing factory. Thus, brownfield planning requires planning engineers to consider the existing conditions and potential constraints of the factory. As such, the degree of freedom is smaller than in greenfield planning.

Both planning entirely new factories as well as restructuring existing ones usually involve a conceptual planning phase which is followed by a detailed planning phase. In the conceptual planning phase, planning engineers aim to generate different layouts without considering every detail. Thus, the main effort is generated by placing functional units (e.g., machines or warehouses) under consideration of specific planning objectives. Typically, not all planning objectives can be achieved simultaneously. Hence, prioritizing different objectives results in a set of different layouts. In this step, automated planning approaches can support the planning engineers in generating concept layouts. Afterwards, the generated optimized concept layouts need to be reviewed and potentially adjusted through human intervention. The best layouts are then transferred to the detailed FLP phase where planning engineers refine the layout within each unit. This includes, for example,

adjusting the position and orientation of single components (e.g., workstations or storage racks) to enhance ergonomic aspects and reduce further non-value-adding activities like walking distances or searching times.

Previous research deemed the application of XR technologies for FLP supportive [50, 28] and explored the application of MR [146, 72, 7, 95] and VR [123, 9, 64] using both HHDs [146, 95] as well as HMDs [146, 123, 72, 9, 7, 64].

Visualizing the planned modifications of a factory in a realistic 3D format, can support their evaluation as well as the communication among stakeholders as presented by Nåfors et al [123]. Consequently, issues caused by the modifications can be identified prior to their actual installation and reduce the need for re-planning. In conventional FLP, the incomplete digital documentation of a factory's physical constraints can impede the evaluation by the stakeholders. In this context, the application of MR is promising since it allows to seamlessly integrate virtual models of missing factory components into the physical scene. In this way, the proposed factory design can be evaluated under consideration of the exact local conditions like presented by Kokkas and Vosniakos [95]. Thereby, simulations of the production process can be added as virtual augmentations.

Alternatively, an existing factory can be virtually replicated, for example through 3D laser scans, and augmented with CAD models of new equipment such as presented by Gong et al. [64]. Similarly, 3D laser scans of an existing factory were integrated with CAD data by Nåfors et al. [123]. In this way, the created virtual environment displays the new layout proposal along with simulations of workers and forklift traffic movements. Similar approaches could also be used to enable remote FLP (i.e., planning engineers who are physically distant from the real factory can review changes from a distance).

MR-HHDs and MR-HMDs were further considered for the validation of new intra-logistics designs by Rohacz et al. [146]. Moreover, Herr et al. [72] presented a MR-HMD application for adapting and comparing layouts with integrated production simulation. An example for collaborative planning using MR-HMDs which integrate production simulation was presented by Baroroh and Chu [7]. Thereby, one user can arrange the components of the production system to construct the factory while the other user interacts with both physical and virtual objects to perform manual tasks of the production process.

6.2 Robot Control

Robotic systems are currently being developed for various applications ranging from healthcare to space exploration. Thereby, different levels of autonomy are being considered – from working independently to collaborating with and being controlled by human operators. In the context of XR^S, the integration of a robotic system was proposed to handle the manipulation of distant, large, heavy, or hazardous objects. To ensure optimal human-robot cooperation, appropriate UIs are required.

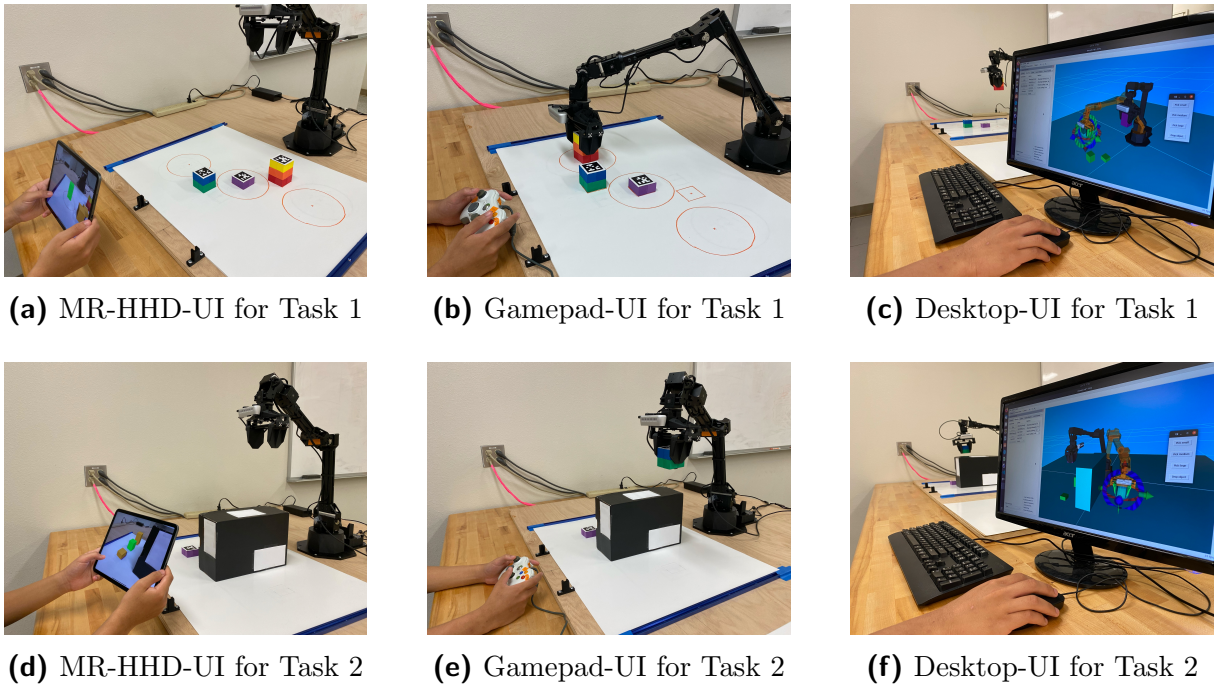
In this context, applying ubiquitous tools like tablets as MR-HHDs seems promising. Previous research on MR-HHDs for robot control such as presented in Chapter 6.1.1, however, mainly employs touch input which requires holding the HHD with one hand and is thus deemed unfavorable. As explained in Chapter 5.1, the HCI community has proposed device-based interaction methods as an alternative which maps the HHD's movement to the virtual objects being manipulated. Yet, such device-based interaction has not been applied for human-robot interaction.

A very detailed survey on the integration of virtual augmentations in the context of robotics was presented by Suzuki et al. [164]. They reviewed related research regarding several criteria. These include (1) the approach used to augment reality in robotics (i.e., concerning hardware and location of virtual augmentations), (2) the augmented robot's characteristics, (3) the virtual augmentation's purpose and benefits, (4) the type of information provided, (5) the way in which this information is presented (i.e., design components), (6) the level of interactivity and interaction modalities, (7) domains for application, and (8) evaluation strategies.

In the context of this taxonomy, our MR-HHD-UI for robot control can be described as follows: The UI (1) uses an HHD to display virtual augmentations (an approach which was only used by 6% of the papers they reviewed), it is used to control a (2) tabletop-size robotic arm which is operated by a single user who is co-located (i.e., at the same site) with the robotic system, and supports (3) robot control through (4) virtual augmentations which first display the object's current and after manipulation its target location. Furthermore, (5) spatial references and visualizations are used to display points and locations combined with virtual replicas of scene objects. The UI can be (7) applied in any domain which involves pick and place operations and was (8) evaluated using a comparative study. However, the (6) interaction paradigm (i.e., in our case device-based object manipulation) cannot be classified with their taxonomy, emphasizing the relevance

of the research gap addressed.

Apart from the newly developed MR-HHD-UI, a Gamepad-UI and a Desktop-UI (see Fig. 6.1) were implemented and evaluated in a comparative study. In the following, descriptions of the implementation of the three UIs, the experimental setup, and the results of the comparative study are provided.



(a) MR-HHD-UI for Task 1 **(b)** Gamepad-UI for Task 1 **(c)** Desktop-UI for Task 1

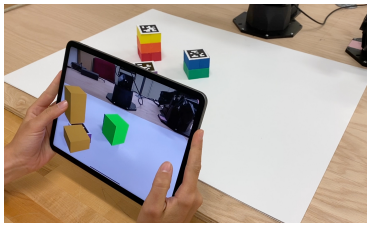
(d) MR-HHD-UI for Task 2 **(e)** Gamepad-UI for Task 2 **(f)** Desktop-UI for Task 2

Figure 6.1: Overview of the developed UIs for controlling the robot arm.

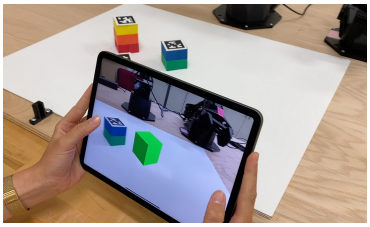
6.2.1 User Interfaces

The development and evaluation of the UIs were part of a joint project between the HCI lab at RPTU and the Department of Mechanical and Aerospace Engineering at the University of California – Davis (UCD). The MR-HHD application for manipulating virtual objects was developed at RPTU while the implementation of the robot applications was carried out by UCD. The interface connecting the MR-HHD application with the robot application was developed in collaboration. The control UIs which allow operating the robot via Gamepad or Desktop were designed jointly with UCD.

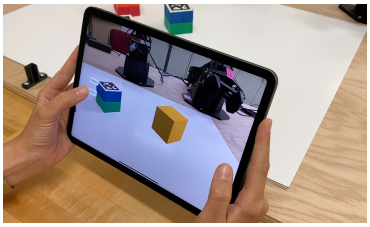
The MR-HHD-UI implements the basic interaction paradigm for translation introduced in Chapter 5.3.2. The steps involved to instruct the robot arm are displayed in Fig. 6.2. Objects can be selected by centering them in the HHD's screen. If this is the case, the



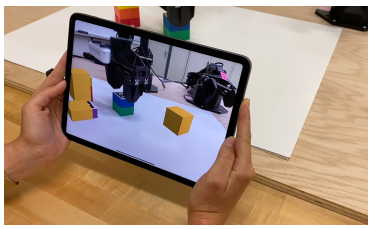
(a) center object in the screen and then apply left-thumb-touch



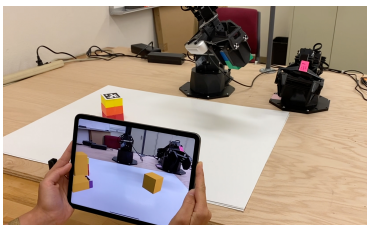
(b) move the tablet to map tablet movement to the virtual object



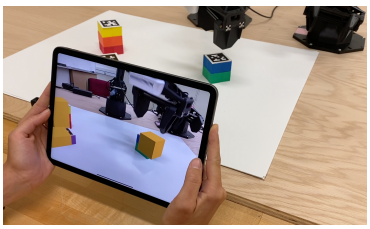
(c) release touch and press the confirm button



(d) robot picks the corresponding physical object



(e) robot moves the physical object



(f) robot drops the physical object at the position of the virtual object

Figure 6.2: Using the MR-HHD-UI to command the robot arm to pick and place an object.

object changes its color and can be translated by applying left-thumb-touch while moving the device. The object is then translated by mapping the HHD's translation in space to the object. To this end, the virtual object's position is updated in every frame by adding the translation vector of the HHD's movement as long as touch input is registered on the left side. When touch is released again and the object is outside the screen's center it is unselected. After the object is unselected, the user can click the confirm or repeat button which appears on the screen's left side. After clicking confirm, the object's id and new position are shared with the robotic system which commands the robot arm to pick up the desired physical object and places it in the new position.

The Gamepad-UI is based on an Xbox 360 controller with labeled buttons. The controller was connected to the robot computer. The controller's left joystick can be used to move the robot's end effector to the left, right, back, and front in real time. To move the robot arm up and down, the controller's left and right trigger can be used. The gripper can be opened and closed via buttons on the controller's right side.

The Desktop-UI for controlling the robot consisted of a custom Python GUI and RViz windows which display a 3D model of the robot arm. The user can command the robot to pick up an object by clicking a button in the GUI. While this step is semi-automated,

placing the object in a new position requires the user to adjust the robot's end effector by manipulating the respective arrows (up/down, left/right, back/front) and circles (yaw, pitch, roll) in RViz. At any time, the user can click a button to plan and execute robot movement such that the real robot moves to the specified target position. Once the robot is in the desired position, the gripper can be opened via button click as well.

Depending on the UI, different frameworks and libraries were used to control the robot's movement and perception. In all settings we used the 6 DOF robot arm, ViperX 300, running ROS Noetic on an Ubuntu PC. ROS packages from Interbotix were used to control the arm. They include motor drivers, 3D and inertial models, and gamepad teleoperation support for the arm. Furthermore, the framework MoveIt was used to perform motion planning of the robot arm while using the MR-HHD-UI and Desktop-UI. For the Gamepad-UI, Interbotix's ROS package for reading gamepad inputs is used to map input via the Xbox 360 controller to the respective robot commands for controlling the arm and the gripper.

Both the MR-HHD-UI and the Desktop-UI require scene capturing. To this end, an Intel RealSense D435i camera was attached to the robot arm. The camera detects the pose of the differently sized foam boxes equipped with AprilTag markers which were placed in the scene (70cm $\times$ 60cm). The images from the camera were collected using the Intel RealSense ROS wrapper and further processed with the ROS package apriltag_ros. In this way, objects could be identified via their unique April-Tag and pose detection can be performed. When objects are detected with their AprilTag, they are added to the planning scene in MoveIt.

The Desktop-UI then allows the user to command the robot to grasp and release objects via button clicks. Then, MoveIt plans and executes the trajectory for the robot to complete the action and executes Interbotix's motor controllers. The object can be moved to a new position by adjusting the pose of the robot's model in RViz accordingly. Through button-click the real robot is then commanded to the specified pose.

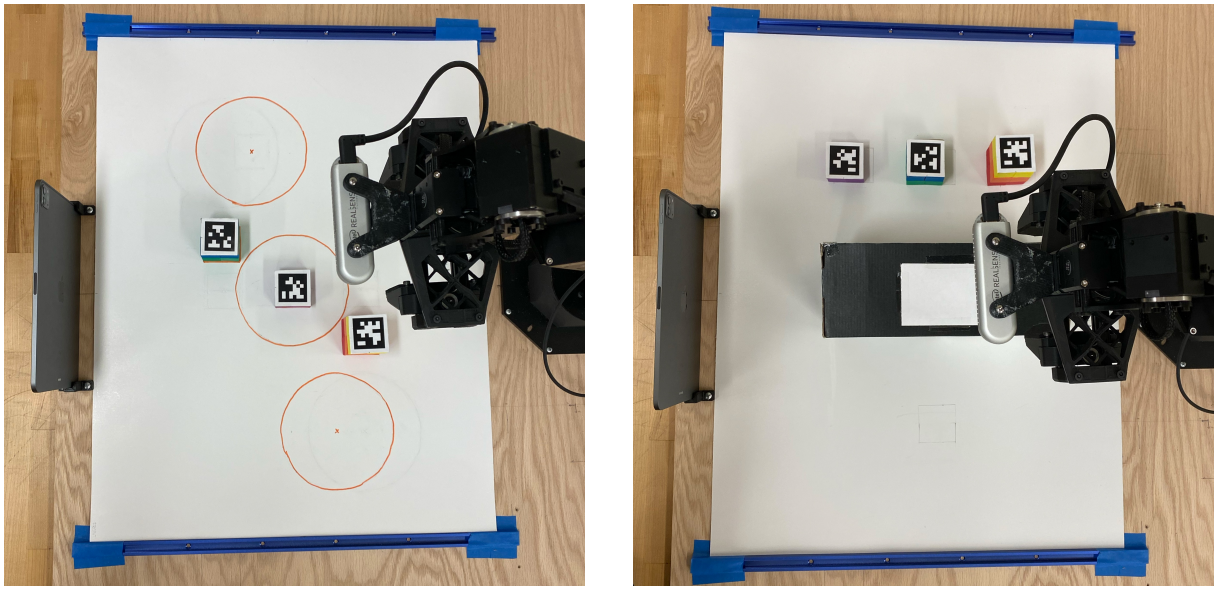
The MR-HHD application runs independently on an Apple iPad Pro (11 inch, Gen. 3) and was developed in Unity using ARKit within the AR Foundation package. In contrast to the other two UIs, this setup requires wireless communication between the MR-HHD and the robot computer. To this end we used Flask, a Python web framework, on the robot's side to listen and respond to the UnityWebRequests from the MR-HHD. Virtual replicas of the foam boxes were added to the scene in the MR-HHD-UI and named according to the identifiers used by the robot application. Upon launch the MR-HHD application

requests the robot application to share the objects' size, position, and orientation. The robot application responds with the requested data. Since the robot and the MR-HHD applications use different coordinate systems, the MR-HHD was always started in a fixed position such that the scene object's positions in the robot application's coordinate system could be transformed to the MR-HHD application's coordinate system based on the fixed offset between the MR-HHD's starting position and the origin of the robot application's coordinate system. The MR-HHD application processes the data obtained from the robot application to adapt the virtual scene objects' sizes, orientations, and positions. Upon the completion of this calibration phase, the MR-HHD can be moved independently in space to select and translate objects to control the robot. As soon as the translation of a virtual object is confirmed, the MR-HHD sends the respective object's id and its target position to the robot application which then transforms the target position back to the respective position in the robot's coordinate system and executes the task using MoveIt's motion planning software and Interbotix's hardware controllers.

6.2.2 Experimental Design, Tasks, and Procedure

The user study was conducted with 20 participants (13 male / 7 female, 14-59 years old) who had different levels of prior experiences with MR, robotics, gamepads, and HHDs. We followed a within-subject design such that all participants completed the tasks with all three UIs. To avoid learning effects, the participants were assigned different starting conditions. They either started with input via MR-HHD or non-MR-HHD. Concerning non-MR-HHD input, we further randomized the order of the Gamepad-UI and the Desktop-UI.

Before a participant completed the tasks with one of the UIs a video tutorial explaining the respective interaction technique was shown. The participants could replay the tutorial as often as they wanted. Then, the participant completed Task 1 and Task 2 with the respective UI. In Task 1 (see Figs. 6.3a, 6.1a, 6.1b, and 6.1c) the participants were asked to sort objects according to their size. In fact, three foam boxes with different heights (small: 2.5cm, medium: 5cm, large: 7.5cm; all: width = length = 5cm) had to be sorted from large to small (front to back). Initially, the boxes were placed in the following order: medium, small, large (front to back). Thereby, the target positions were marked as circles. The participants had to translate the large (Task 1a), small (Task 1b), and medium (Task 1c) object to their target position. The distances from start to target positions were 31.5cm for the large, 12cm for the medium, and 21cm for the small object.



(a) Task 1 **(b)** Task 2

Figure 6.3: Task setup during MR-HHD application launch.

A task was successfully completed when the foam box was placed upright and touched the target area. For Task 2 (see Figs. 6.3b, 6.1d, 6.1e, and 6.1f) the participants had to move the medium sized box to the right side of an obstacle box (28cm × 10.5cm × 18.5cm). Thereby, all objects were initially placed on the left side of the obstacle box and the task was successfully completed if the medium sized box was placed upright on the right side of the obstacle box. To evaluate and compare the three UIs we logged if tasks were completed successfully and measured task completion times (TCTs) throughout the experiments. After the completion of all tasks with one UI, the participants provided feedback through the NASA TLX and rated their agreement with statements regarding the interaction technique's suitability for the task, conformity with their expectations, learnability, and satisfaction. This procedure was then repeated for all three UIs. At the end of the experiment, the participants were asked to rank the three UIs according to their preference.

6.2.3 Results

We compared the effectiveness of the UIs based on the percentage of successfully completed tasks. While available MR technologies are known to provide insufficient real-time tracking [164] our work was focused on applying an advanced object manipulation method for MR-HHDs to robot control rather than on improving display and tracking technologies.

Therefore, a task was considered to be successfully completed when the corresponding foam box was placed upright within its target area (radius: 8cm). As shown in Fig. 6.4, 96% of all tasks were completed successfully using the MR-HHD-UI and Gamepad-UI whereas only 76% of the tasks were completed successfully with the Desktop-UI.

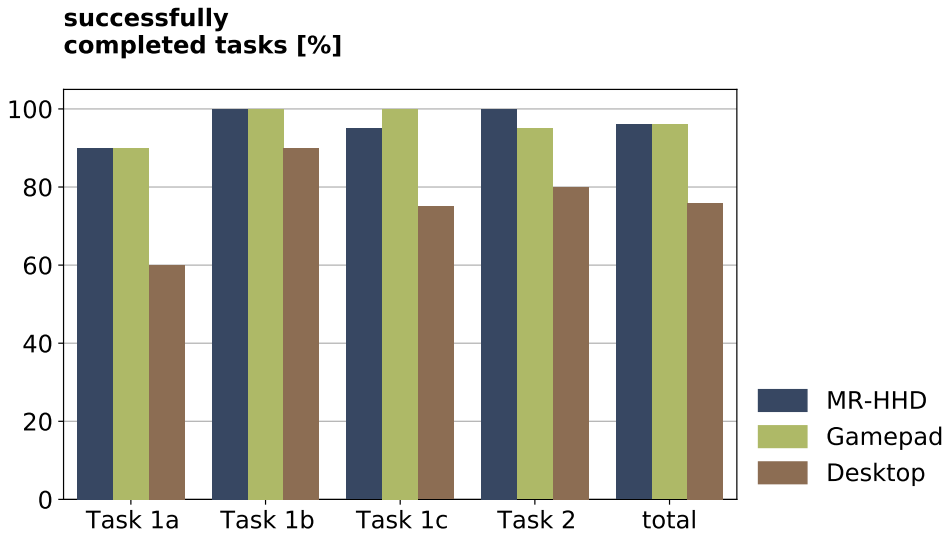


Figure 6.4: Effectiveness of all UIs assessed through success rates.

The efficiency of the UIs was compared based on TCTs of successfully completed tasks. In this regard, it is relevant to consider the different workflows of the three UIs. When using the MR-HHD-UI, input is provided before the robot executes the commands. With the Gamepad-UI user input and robot execution occur in parallel and with the Desktop-UI user input and robot execution occur alternately. To maintain comparability, when analyzing TCTs, we considered the time spans during which user input was required. For each task and UI we measured two different TCTs: the time needed to instruct the robot to translate the object with selecting the object ($TCT_{w/\_sel}$) and without selecting the object (TCT_{w/o_sel}). In addition, we considered two different $TCTs_{w/\_sel}$ for the MR-HHD-UI: the time needed to instruct the robot (MR-HHD_instr) and the total time for user input and robot execution (MR-HHD_total).

The mean $TCTs_{w/o\_sel}$ (see Fig. 6.5) and mean $TCTs_{w/\_sel}$ (see Fig. 6.6) for all successfully completed tasks and UIs indicate that both the MR-HHD-UI and Gamepad-UI clearly outperformed the Desktop-UI. Furthermore, Fig. 6.5 and Fig. 6.6 show that regarding the time needed to instruct the robot the MR-HHD-UI achieved the best average performance. Paired samples t-tests with Bonferroni correction showed significantly lower $TCTs_{w/o\_sel}$ of Tasks 1b ($p \leq .01$), 1c ($p \leq .05$), and 2 ($p \leq .05$) for the MR-HHD-UI compared to the Gamepad-UI (alternative hypothesis $\mu TCT_{w/o\_sel}(\text{MR-HHD}) < \mu TCT_{w/o\_sel}(\text{Gamepad})$)

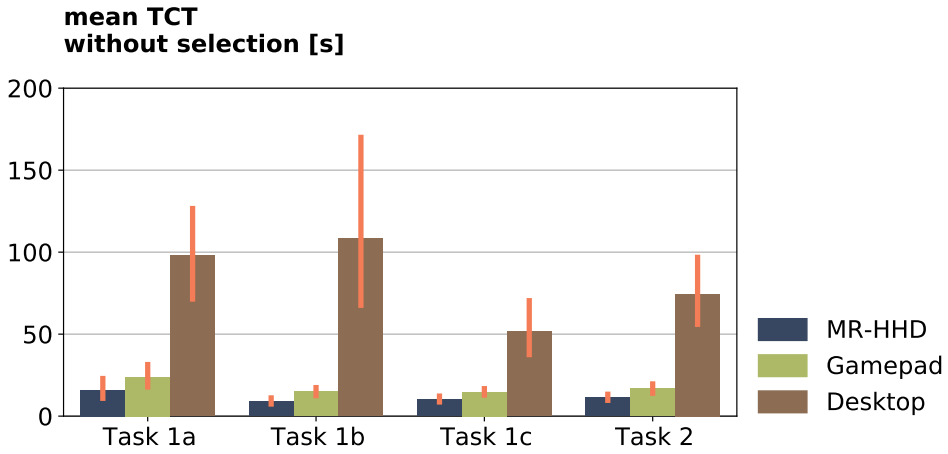


Figure 6.5: Task completion without selection: Mean TCTs and 95% confidence intervals for successfully completed tasks.

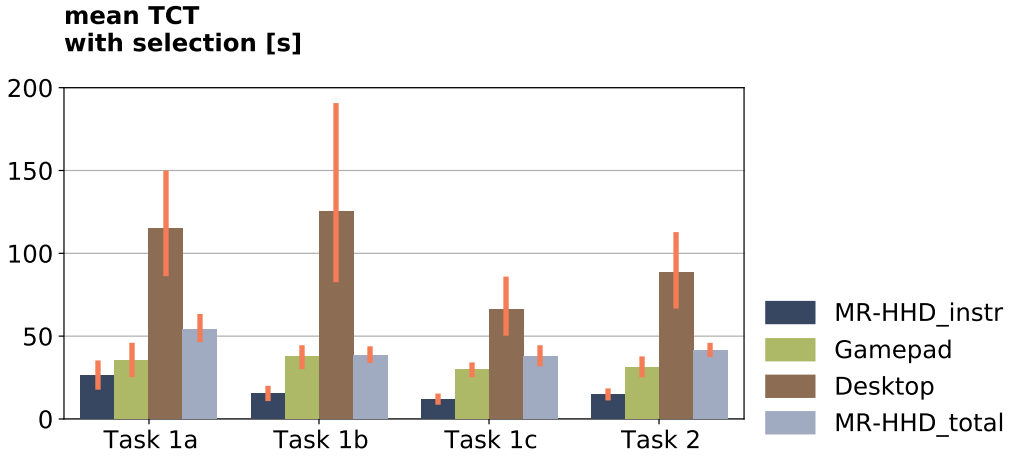


Figure 6.6: Task completion with selection: Mean TCTs and 95% confidence intervals for successfully completed tasks.

and significantly lower $\mathrm{TCT}_{\mathrm{w/\_sel}}$ of Tasks 1b ($p \leq .0001$), 1c ($p \leq .0001$), and 2 ($p \leq .0001$) for instruction with the MR-HHD-UI compared to the Gamepad-UI (alternative hypothesis $\mu\mathrm{TCT}_{\mathrm{w/\_sel}}(\mathrm{MR\text{-}HHD\_instr}) < \mu\mathrm{TCT}_{\mathrm{w/\_sel}}(\mathrm{Gamepad})$).

While the mean $\mathrm{TCTs}_{\mathrm{w/\_sel}}$ for MR-HHD_total were higher than for the Gamepad-UI, they are still lower than the corresponding time needed with the Desktop-UI (see Fig. 6.6). Furthermore, the robot moved at rather low speed while executing the commands from the MR-HHD. MR-HHD_total could therefore be easily reduced by accelerating the speed at which the robot moves. As the study was mainly focused on comparing the usability of different UIs for robot control, we consider the MR-HHD_instr to be more representative than MR-HHD_total for comparing $\mathrm{TCTs}_{\mathrm{w/\_sel}}$.

After task completion with one UI, the participants rated their agreement with seven statements about the interaction technique's suitability for the task, conformity with their expectations, learnability, and satisfaction (see Fig. 6.7). Similar to the results regarding effectiveness and efficiency, the Desktop-UI was rated worst and the MR-HHD-UI and Gamepad-UI received similar higher ratings. The overall mean rating for the Gamepad-UI (4.41) was slightly better than for the MR-HHD-UI (4.29). When computing the mean ratings for single statements the MR-HHD-UI was rated slightly better regarding the number of steps to be performed for task completion and the ease with which the technique can be relearned after a lengthy interruption. After task completion with all three UIs, the participants were asked to rank the UIs using scores from 1 (best) to 3 (worst). Here, the mean scores show that the MR-HHD-UI was rated as the favorite UI (1.45), followed by the Gamepad-UI (1.7), and Desktop-UI (2.85).

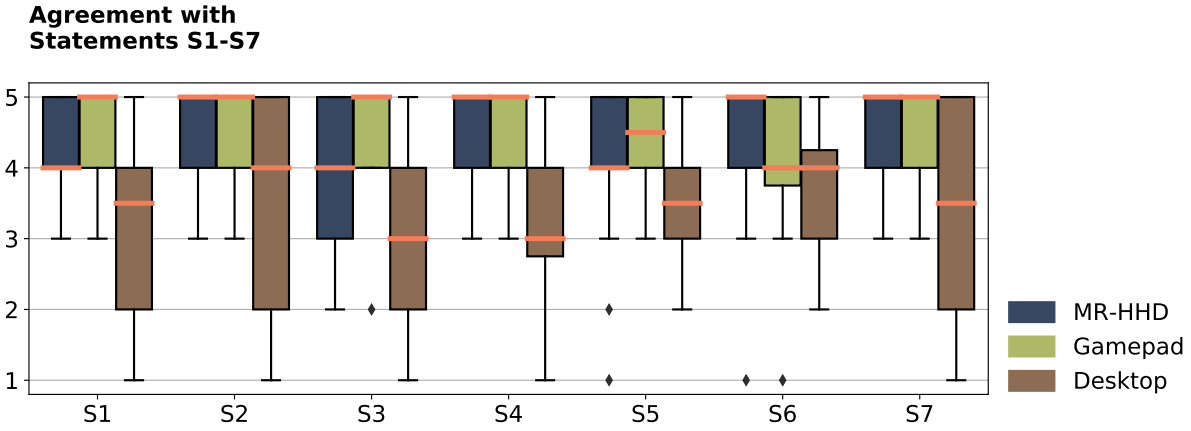


Figure 6.7: Participants' agreement with statements S1 — S7 from 1 (predominantly disagree) to 5 (predominantly agree) regarding all UIs: **S1** The interaction technique is well suited to the requirements of the task; **S2** The number of steps to complete a task with the interaction technique is adequate; **S3** When instructing the robot with this interaction technique, I have the feeling that the results are predictable; **S4** It was very easy for me to learn how to use the interaction technique; **S5** I needed a short time to learn the interaction technique. **S6** Relearning the interaction technique after a lengthy interruption will be easy; **S7** Overall, I am satisfied with this interaction technique.

For every UI, the participants answered the NASA TLX questionnaire and the weighted ratings were computed according to [125]. The total weighted workload as well as the weighted ratings for each subscale are shown in Fig. 6.8. The total weighted workload for the MR-HHD-UI (25.22) was slightly lower than for the Gamepad-UI (28.06) and the highest workload was experienced while using the Desktop-UI. Considering the meta-analysis of NASA TLX scores from Grier [70], the MR-HHD-UI resulted in a workload which is lower than in at least 90% of the studies reviewed while the Gamepad-UI's

workload is only lower than in at least 80% of the studies reviewed. Fig 6.8 further shows that the MR-HHD-UI evoked less mental demand and effort than the Gamepad-UI. Concerning the experienced physical demand, the MR-HHD-UI received higher ratings than the Gamepad-UI. This seems reasonable as the HHD has to be moved in space while instructing the robot.

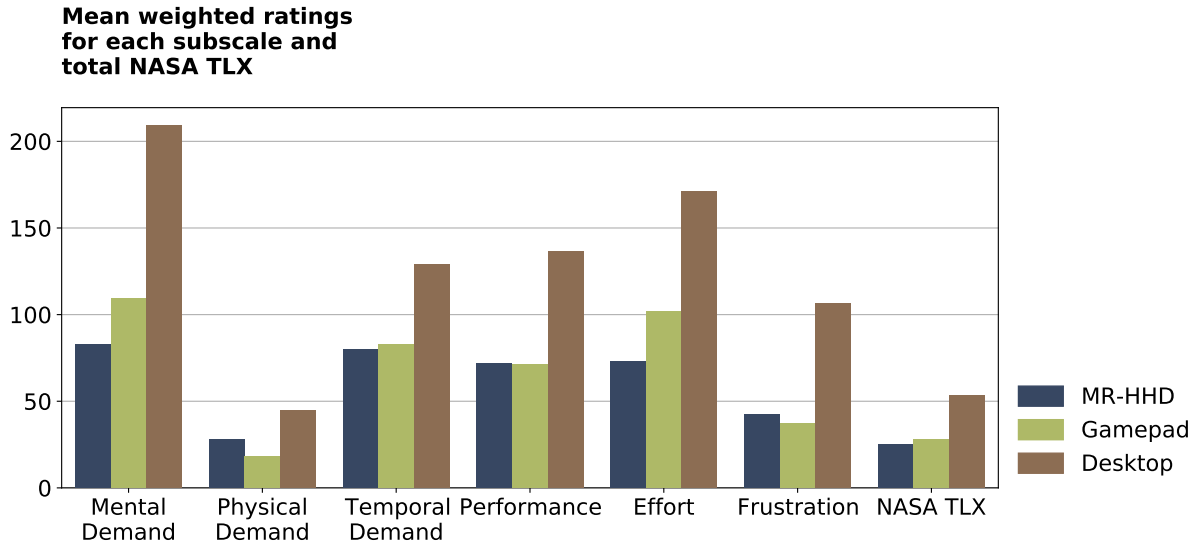


Figure 6.8: Mean weighted NASA TLX ratings.

To further compare the MR-HHD-UI and the Gamepad-UI, we took the participants' prior experience into account. Thus, we conducted paired samples t-tests with Bonferroni correction (alternative hypotheses $\mu\text{TCT}_{\text{w/o\_sel}}(\text{MR-HHD}) < \mu\text{TCT}_{\text{w/o\_sel}}(\text{Gamepad})$ and $\mu\text{TCT}_{\text{w/\_sel}}(\text{MR-HHD\_instr}) < \mu\text{TCT}_{\text{w/\_sel}}(\text{Gamepad})$) for groups of participants without previous Robotics-experience (n=14), MR-experience (n=16), and Gamepad-experience (n=5). Thereby, the $\text{TCTs}_{\text{w/o\_sel}}$ were significantly lower using the MR-HHD-UI than using the Gamepad-UI for participants without MR-experience (Task 1b: $p \leq .01$; Task 2: $p \leq .05$), without Robotics-experience (Task 1b: $p \leq .05$), and without Gamepad-experience (Task 1c: $p \leq .05$). Furthermore, the MR-HHD-UI's $\text{TCTs}_{\text{w/\_sel}}$ when instructing the robot were significantly lower than those of the Gamepad-UI for participants without MR-experience (Tasks 1b, 1c, 2: $p \leq .0001$), without Robotics-experience (Tasks 1b, 1c: $p \leq .0001$; Task 2: $p \leq .001$), and without Gamepad-experience (Tasks 1b, 1c: $p \leq .01$).

When comparing mean ratings of the participants' agreement with S1-S7 (see caption of Fig. 6.7), we found that participants without Gamepad-experience rated the MR-HHD-UI (4.6) better than the Gamepad-UI (4.29). As stated above, the mean ratings for single statements indicate slightly better results for MR-HHD-UI than the Gamepad-UI regarding the number of steps that have to be performed and the expected time needed

to relearn the interaction technique. These effects were amplified for participants without experience in robotics or MR.

On top of that, the differences between the workload measured by the NASA TLX for the MR-HHD-UI and Gamepad-UI increased for participants who had no prior experience with robotics or MR. A particularly large discrepancy between the workloads of the MR-HHD-UI (18.6) and the Gamepad-UI (38.4) was observed for participants without prior experience with Gamepads. Based on these insights, we rate the MR-HHD-UI to be a powerful tool for robot control which is particularly well-suited for inexperienced users.

6.3 Factory Layout Planning

Factory layout planning (FLP) represents a particularly promising application area for XR technologies, with relevance across diverse domains. As summarized in Chapter 6.1.2, FLP concerns the building of new factories but also the restructuring of existing factories. This can for example be the case when a new machine needs to be integrated into an existing factory or when the design of a product gets updated such that the shape and size of its processed parts change.

In this context, the application of XR technologies can save both time and costs as they allow evaluating working environments in virtual or virtually augmented environments. Previous research has demonstrated how both MR and VR as well as HHDs and HMDs can support different FLP use cases. The majority of solutions described in Chapter 6.1.2 is focused on providing accurate visualizations and simulations for specific use cases. While some of the presented applications allow layout adaptations, these interaction modalities are either only described to a very limited extent and lack adequate evaluations or they are tailored to very specific use cases. However, planning engineers are often involved in multiple of these planning cases, requiring them to switch between XR applications that involve different technologies. Hence, scalable UIs which support the precise manipulation of factory components across different XR technologies and FLP use cases are highly relevant but still missing. Addressing this gap, we demonstrate how concepts of XR^S can be applied across different planning cases and XR technologies. More specifically, we present the development of UIs for XR-supported FLP which integrate the most suitable technology while providing consistent spatial interaction to facilitate switching between XR technologies and use cases.

We begin with a comprehensive analysis on how XR's potential can be leveraged to address requirements in different phases of single- and multi-user FLP. Building on the results of the analysis, we apply the consistent spatial interaction techniques from Chapter 5 across different XR technologies and FLP phases. This results in a multi-user MR-HHD application for conceptual FLP and a VR-HMD application for detailed FLP which were evaluated in a pilot study. Eventually, we present the results of the pilot study, structured into the observations made, feedback gained in the interviews, and a list of opportunities for future research and extensions of the prototypes.

6.3.1 Analyzing XR's Potential to Support Factory Layout Planning

To analyze the potential of XR we consider a set of use cases within the FLP process which can be described by the combination of different FLP-approaches {greenfield, brownfield}, planning phases {conceptual, detailed}, and user roles {single user, co-located collaborators, remote collaborator(s)}. For conceptual FLP we consider the size of the factory as an additional parameter and for greenfield FLP we take into account whether the physically existing factory components are fixed or manipulable. Based on the following analysis we provide recommendations for choosing XR technologies in different FLP use cases (see Fig. 6.9).

As summarized in previous chapters of this dissertation, HCI researchers have explored the application of XR to support collaboration independent of FLP. In this context, the combination of MR, VR, and different avatars has been proposed (e.g., [5, 138, 140]). Thereby, on-site users can be provided with MR-UIs and off-site users can join through VR-UIs displaying a virtual replica of the local scene.

In the conceptual planning phase generating and comparing layouts requires human intervention and therefore an interactive layout visualization. In this phase, planning engineers need to arrange entire factory units (i.e., adapting the unit's position and orientation) rather than single components of a factory (e.g., a single machine). Therefore, a 2D top view of the factory is in this stage often sufficient. In contrast to usual 2D screens, XR allows visualizing the factory layout as 2D from the top while also allowing easy transitions into 3D. By displaying the 3D model of the factory from the top users are given the option to manipulate factory units in 2D but also to zoom into a unity for inspecting single parts of a unit in 3D. Such top views can be displayed on both HHDs and HMDs. HHDs provide high portability and are already well-established in many workplaces for

non-XR purposes. Thus, they usually offer a more cost-effective and flexible entry point to XR than HMDs. On the other side, however, HHDs offer less screen space than HMDs which makes them less suitable for displaying layouts of large factories. As described above, co-located collaborators may thereby be provided with MR devices and remote collaborators with VR devices. Single users may choose the degree of virtuality. In the conceptual planning phase, both greenfield and brownfield planning start with this 2D arrangement of factory units. During brownfield FLP certain constraints (e.g., the position of a specific unit is fixed) might have to be integrated. Yet, in general, the same application design can be used, leading to the same recommendations regarding the choice of XR technologies.

On the contrary, detailed planning requires more realistic 3D visualizations. Here, HMDs allow accessing single factory units in real size which can facilitate the evaluation of ergonomic aspects and distances. On top of that, the higher immersion offered by HMDs (in comparison to HHDs) allows simulating working processes within different arrangements of the unit's components. Thereby, the ideal degree of virtuality depends on both the FLP approach (i.e., greenfield vs. brownfield) and location of the users. In the case of detailed greenfield FLP, VR-HMDs can offer large and highly immersive environments for single or remote planning engineers. Co-located collaborators can benefit from MR-HMDs which offer less immersion but maintain natural communication. Brownfield planning might require the consideration of physically existing components whose position and orientation is fixed and should not be manipulable during the planning session. In this context, MR-HMDs seem promising as they allow seamlessly integrating virtual (i.e., not yet existing) factory components into the physical factory. In this way planning engineers who are located on site can arrange the virtual components in relation to the fixed existing components. These on-site planning engineers can then be joined by remote collaborators through VR-HMDs which display a virtual replica of the on-site environment. Other brownfield FLP use cases may require re-arranging existing components. Here, VR-HMDs may be more suitable – even for co-located collaborators – as VR allows them to manipulate virtual replicas of physically existing factory components. In summary, the following conclusions can be drawn regarding the potential of different degrees of virtuality and devices:

- **HHDs** offer highly accessible, portable, and cheap entry points to XR. They are deemed particularly useful for the conceptual planning phase as they allow interacting with 2D top views of the factory while also enabling transitions into 3D.

- **HMDs** offer more screen space than HHDs which makes them a promising alternative not only for displaying layouts of large factories in the conceptual planning phase but also for displaying real size factories in 3D during the detailed planning phase.

- **MR** allows to seamlessly integrate real and virtual factory components. In this way, it can provide co-located collaborators with virtual content while maintaining natural communication. During detailed brownfield FLP, MR further allows to virtually augment physically existing factory components whose position and orientation is not supposed to change.

- **VR** offers detailed immersive and interactive visualizations which can provide virtual replicas of physically existing factories for single or remote users during detailed FLP. On top of that, VR can support detailed brownfield planning which involves re-arranging physically existing components.

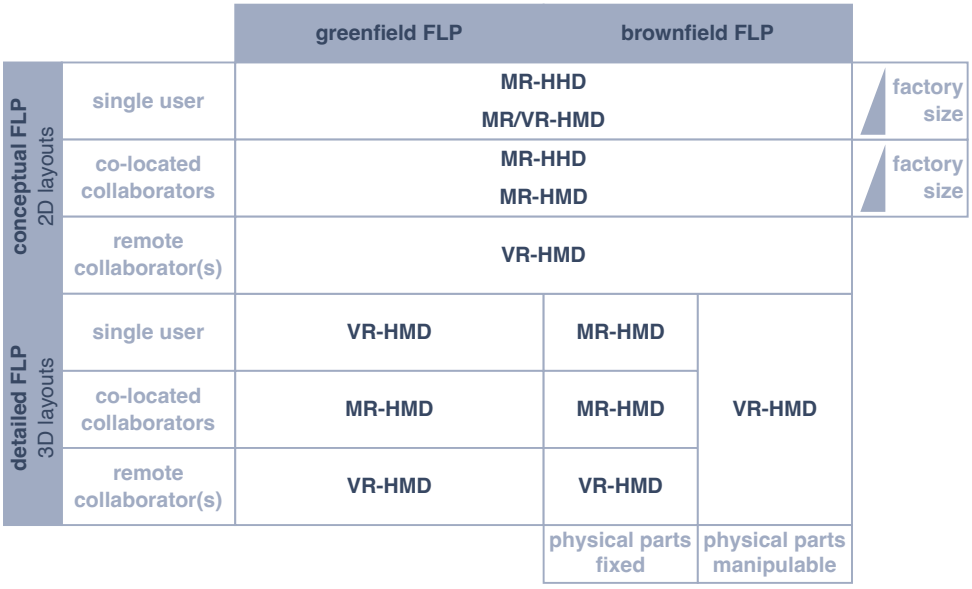


Figure 6.9: Recommended XR technologies for the respective FLP use cases. In the use cases of detailed brownfield planning, it is assumed that the single user and the co-located collaborators are located at the existing factory; any user who is not located at the existing factory is in this case treated as a remote collaborator.

Planning engineers are likely to be involved in multiple of these use cases as their location may change and the FLP process progresses. For instance, a user's role may change from a co-located collaborator to a single user who is located on site to a remote collaborator. At the same time, a planning engineer can be involved in several greenfield and/or brownfield

FLP processes which are in different planning phases (i.e., conceptual or detailed). Consequently, using the most suitable XR technology (see Fig. 6.9) in every case will require switching between technologies. Thus, consistent UIs are required which were identified as absent in published work on XR-supported FLP in Chapter 6.1.2. Making switching among different XR technologies as seamless as possible is the fundamental purpose of the XR^S concept introduced in Chapter 3 and the UIs presented in this dissertation. To investigate their applicability to FLP, we extend and evaluate the spatial interaction methods presented in Chapter 5 to two FLP use cases.

6.3.2 User Interfaces

Building up on the conclusions drawn from the analysis and to cover a broad spectrum of both XR technologies and FLP use cases, we developed two prototypes in Unity using ARKit and XR Interaction Toolkit: a collaborative MR-HHD application and a VR-HMD application.

The MR-HHD application (see Figs., 6.10, 6.11, and 6.12) allows planning engineers to rearrange factory units during conceptual FLP. The application can be used by single or multiple users who can manipulate the positions and orientations of factory units simultaneously using their own MR-HHD. In parallel, they can monitor the effects of their manipulations on transportation intensity in real time. A menu thereby allows to save and select different layouts.

With the VR-HMD application (see Figs. 6.13 and 6.14) a planning engineer can refine single factory units during detailed FLP. Our application allows adjusting the position and orientation of machines, workstations, racks, and working material to reduce walking distances and enhance ergonomics.

Both applications use the same factory and spatial interaction designs which we describe below. Following this, we explain how these designs were implemented in the MR-HHD and VR-HMD application.

6.3.2.1 Factory Design

The virtual factory model was created by combining our custom models with free models from [53, 173]. The factory consists of the following six units: A warehouse unit consisting

of storage racks; three machine units including both storage racks and machines; a quality control unit consisting of automated guided vehicles, forklifts, and the control station; and a working station unit used for commissioning which consists of a workstation, storage racks, and an assembly line setup with conveyor belt. The models inside a unit are assigned as children of the unit and the factory units are assigned as children to a plane in a hierarchical setting. In the MR-HHD application the plane containing all factory units is further surrounded by a drop-off area.

6.3.2.2 Spatial Interaction Design

To develop UIs which stay consistent when planning engineers have to switch between the use cases and applications, we first identify operations which are required in both use cases and discuss how *Move'n'Hold* as presented in Chapter 5 is extended and implemented accordingly.

Selecting Objects. Both use cases require the selection of different virtual objects. During conceptual planning with the MR-HHD application users need to be able to select entire factory units and during detailed planning with the VR-HMD application users need to be able to select virtual objects inside a unit such as machines, racks, working material, or workstations. Thereby, the selection of a virtual object should also include the selection of its children. Both of our applications use the selection paradigm presented in Chapter 5.4. To this end, a red point is added to the center of the HHD/HMD view and the applications constantly check if a ray emerging from the MR-HHD/VR-HMD hits a manipulable object in the scene. Upon the detection of a hit, the selection point disappears and the selected object's color is adapted as a highlight mechanism.

Translating and Rotating Objects. After selecting a virtual object, a user should be able to manipulate (i.e., translate or rotate) the respective object. To this end *Move'n'Hold* is employed as follows. In the MR-HHD application users can apply touch with the left thumb anywhere on the HHD's left display side and then translate/rotate the HHD in space. As long as touch input is registered only on the left side, the HHD's translation/rotation in space is mapped to the selected factory unit. By adding touch with the right thumb on the right display side, users can evoke automated continuous manipulations of the selected factory unit. As long as

touch is registered on both display sides, the factory unit will be translated/rotated automatically in the direction of the initial movement which was performed while only left-thumb-touch was active. In this way, users can seamlessly combine small precise and large coarse movements of the factory units. The VR-HMD application implements *Move'n'Hold* for HMDs such that the components of the factory unit seen through the HMD can be manipulated with a tablet controller which implements the same interaction paradigm (i.e., combinations of peripheral touch and device movements) as the MR-HHD application. In Chapter 6.3.2.3 and Chapter 6.3.2.4 we describe how *Move'n'Hold* is adapted to meet the use case specific requirements of the conceptual and detailed planning phase while leveraging the benefits of a consistent interaction paradigm.

Changing the Manipulation Mode. Both conceptual planning as well as detailed planning require translating and rotating objects. The separation of translation and rotation as provided in *Move'n'Hold* thus requires a method to toggle between the translation and rotation mode. To enable seamless combinations of translation and rotation, we propose two toggle methods: the user can either (1) perform a double tap anywhere on the right display side while no object is selected and no touch is performed on the left side or the user can (2) simultaneously apply touch with three fingers anywhere on the display.

6.3.2.3 MR-HHD-UI

The MR-HHD application through which the factory layout can be collaboratively adapted runs on two Apple iPad Pros (11 inch, Gen. 3/4). Networking is thereby realized through Netcode for GameObjects. Our setting involves two users in which one acts as a host (i.e., server and client) while the other user only acts as a client. When the application is launched, the factory is anchored in the physical space. Both users can then simultaneously manipulate factory units as shown in Fig. 6.10.

The positions and orientations of the factory units are thereby constantly synchronized among the devices of both users and *Move'n'Hold* is implemented as follows. As long as no touch is registered, the application constantly checks if a ray emerging from the device center hits a factory unit. If this is the case, ownership for the respective unit is requested.

Figure 6.10: Conceptual FLP with the multi-user MR-HHD app: A factory layout is collaboratively optimized by manipulating factory units using *Move'n'Hold*: The left user uses translation through direct mapping (left-thumb-touch only) whereas the right user applies continuous rotation (left-thumb-touch and right-thumb-touch).

When ownership is permitted (i.e., the unit is currently not being manipulated by the other user), the selection point in the screen's center is disabled and the unit's color is adapted according to its new owner on the devices of all users. Then, the unit can be manipulated by combining peripheral touch and tablet movement. For this, we constantly register the vector/quaternion describing the device's movement in space. During conceptual planning, factory units are supposed to be translated and rotated only in the xz-plane. Therefore, we remove translations on the y-axis as well as rotations around both the x-axis and the z-axis before mapping the vector/quaternion to the unit being manipulated. Apart from this, we implemented a collision prevention feature to ensure that units cannot pass through each other.

Depending on the option chosen for switching from translation to rotation mode and vice versa, the application constantly checks if the user performed a double tap on the right display side or triple touch. Upon the detection of the respective gesture, the manipulation mode is changed internally and the text field showing the active manipulation mode on the top right is adapted accordingly.

Figure 6.11: Conceptual FLP with the multi-user MR-HHD app: To compare the currently active layout against other layouts, the host opens the menu by clicking the respective button.

Our application allows the planning engineers to compare a set of potential start layouts. To do so, the host user can open a menu on all devices by clicking the respective button on the top left (see Fig. 6.11). The menu displays a list of layouts along with their transportation intensity – a key planning KPI which is defined as the sum of the number of transports multiplied by the corresponding transport distance. Thus, lower transportation intensities lead to lower transportation costs, which contribute substantially to a plant's operating costs.

The menu allows the users to collaboratively review and compare the different layouts (see Fig. 6.12). When the host selects a layout, the menu is closed and the chosen layout is displayed on all devices. While the users optimize the factory units' arrangement, the transportation intensity (i.e., the improvement of the transportation costs) is updated in real time and displayed at the top center of the screen. At any time, the host can save a layout by replacing the layout with the worst transportation intensity in the menu (see Fig. 6.12)

Figure 6.12: Conceptual planning with the multi-user MR-HHD app: The currently active layout is saved by replacing the layout with the worst transportation intensity.

6.3.2.4 VR-HMD-UI

The VR-HMD application was developed for the HTC VIVE Pro in combination with a tablet controller (Apple iPad Pro, 11 inch, Gen. 3). As described in Chapter 5.4 the tablet controller shares the registered touch input on the left and right display sides as well as its movements with the VR-HMD application through UnityWebRequests. While the same factory model is used for the VR-HMD application as for the MR-HHD application, a relevant difference between the two apps concerns the interaction with the virtual objects. In contrast to the MR-HHD application, the parent-child relation of virtual objects can change during runtime in the VR-HMD application, for example if working material is moved inside or outside a rack. Consequently, a more flexible implementation is required.

As long as the tablet controller does not register touch input, the VR-HMD application checks if a ray emerging from the HMD (see Fig. 6.13) hits a manipulable object.

If this is the case, the selection point disappears and the object hit by the ray as well as its children are selected and their color is adapted accordingly. The user can then manipulate the selected object (and its children) by moving the tablet while applying touch (see Fig. 6.14). To this end, tablet movement and registered touch events are constantly sent

Figure 6.13: Detailed FLP using the VR-HMD app: An overview of the virtual factory is shown while no component is selected.

to and processed by the VR-HMD application as described in Chapter 5.4.

In line with the MR-HHD application, object rotations in the VR-HMD application are also limited to the y-axis by removing rotations around the x- and z-axis from the tablet's movements before mapping it to the selected components.

Regarding object translation, we follow a different approach compared to the MR-HHD application. Translating objects in three dimensions is enabled and combined with different kinds of automated placement. To investigate how object-specific behavior influences the user experience, we implement different placement mechanisms. Upon release, racks, machines, and workstations drop to the floor whereas the placement of working material boxes depends on the position in which they are released. If they are released inside a rack, they drop onto the closest shelf and if they are released outside the rack, they remain at the position at which they were released.

To maintain consistency with the MR-HHD application, we implemented collision detection to prevent objects from passing through each other and allow users to apply the same touch gesture on the tablet controller for toggling between the translation/rotation mode. To this end, the registered gestures are transferred from the tablet controller to the VR-HMD application through the same pipeline as the device movements and touch events for manipulating objects. Furthermore, the HMD's view contains a similar text field as the MR-HHD application to indicate the currently active manipulation mode.

(a) A rack is translated through direct mapping (left-thumb-touch)

(b) a machine is continuously rotated (left-thumb-touch and right-thumb-touch).

Figure 6.14: Detailed FLP using the VR-HMD app: Factory components are manipulated using *Move'n'Hold*.

6.3.3 Experimental Design, Tasks, and Procedure

The developed prototype applications were evaluated in a pilot study with 6 FLP experts (in three pairwise sessions) who have performed FLP with traditional techniques and 2D desktop-based applications before. In this way, we follow Nielsen's [127] recommendation of performing qualitative usability tests with 5 participants.

In the experiments conducted, the participants were first introduced to the overall procedure and interaction paradigm *Move'n'Hold*. For this purpose, we developed a MR-HHD training application which allows manipulating virtual boxes freely in three dimensions as described in Chapter 5. In contrast to the applications developed specifically for FLP, the training application did not implement manipulation constraints or collision prevention.

A short training phase was provided ahead of the experiments in which the participants were also introduced and asked to choose between the double tap and triple touch for toggling the manipulation modes. After they felt comfortable using *Move'n'Hold*, the participants were introduced to the functionalities of the MR-HHD application and asked to collaboratively optimize the factory layout regarding transportation intensity. This part of the evaluation involved comparing and choosing a layout from the list, selecting and manipulating the factory units to adjust their position and orientation while monitoring the transportation intensity, and saving the optimized layout design. During the experimental session each participant acted as the host once.

Afterwards, the participants continued with the VR-HMD application. Here, each participant optimized a different factory unit by adjusting the position and orientation of its components with *Move'n'Hold* for HMDs while the other participant observed the interaction. The first participant was asked to optimize a working station used for commissioning (see Fig. 6.14a) such that the working material can be transferred to production as efficiently as possible while also considering ergonomic aspects. This part of the experiment required the participant to adjust the position and orientation of several virtual objects. The position and orientation of the racks need to ensure the accessibility of the working material and the boxes storing the working material should be sorted according to their content with heavy items being placed at ergonomic positions. The second participant was asked to optimize the arrangement of virtual objects in a machine unit (see Fig. 6.14b) such that the walking distances while operating a machine are minimized and ergonomic aspects are considered. Thus, the machine's loading point needs to be accessible and close to the most relevant working material stored in the boxes. Similar to the first unit, heavy

items should thereby be stored at ergonomic positions.

After task completion with the MR-HHD and VR-HMD application we conducted a semi-structured interview together with both participants to gain additional feedback on the usability of the applications and the need for any potential improvements.

6.3.4 Results

Both the observations made while the participants interacted with the applications as well as the interviews conducted delivered interesting insights as well as opportunities for future research and extensions which are summarized in the following.

6.3.4.1 Observations

All participants were able to successfully apply *Move'n'Hold* as taught through the training application for manipulating factory units with the MR-HHD application during **conceptual planning**. While we observed that some participants were able to learn and apply the interaction technique faster than others, their performance converged over time.

The participants used Hold (i.e., applying touch on both display sides) to perform large continuous manipulations but also for stepwise translations and rotations when trying to exploit the space most efficiently and to perform small corrections. For these stepwise manipulations, the participants repeatedly applied and released touch on the right side while keeping touch on the left side active. As reported in Chapter 5, the same interaction strategy was also observed in other studies involving *Move'n'Hold*.

In all sessions, equal participation by the users was observed. The task was approached by first getting an overview over the units which involved discussion about their functionality and current arrangement. Then, the participants planned how they intend to manipulate the units and distributed the tasks. Thereby, the participants either referenced units verbally by naming their function (e.g., the warehouse unit) or by hovering over them with the selection point which highlighted the respective unit. Sometimes, the task was split up such that one participant manipulated a unit while the other participant reported the manipulations' effect on the transportation intensity. In our experimental setting, the participants were mostly facing each other which resulted in some difficulties when

referencing directions such as left/right (i.e., the right side of one user is the left side for the other user).

Furthermore, we observed, that the participants did not evoke collisions on purpose. They occurred occasionally when the participants intended to use the space most efficiently and thus moved the units as close as possible to each other. In some cases, participants adjusted the positions of the units first and wanted to rotate them afterwards. Here, rotation was often prevented due to collision prevention as units were already placed very close to each other.

Regarding the options for toggling between translation/rotation mode, all participants preferred the double tap for both the MR-HHD and the VR-HMD application.

When moving on to the **detailed planning** task using the VR-HMD application, the participants were able to intuitively transfer the interaction paradigm *Move'n'Hold* as learned during the training phase and the conceptual planning task with the MR-HHD application.

While the participants were manipulating components of the VR scene, we observed that the tablet controller was held in different positions and orientations (tilted, parallel or orthogonal to the floor, at hip or chest height). The observed equal performance demonstrates *Move'n'Hold*'s suitability to different user preferences.

As already observed for the MR-HHD application, Hold was also used for both large and stepwise object manipulations with the VR-HMD application. Hold was often applied to move a rack closer to the camera in order to improve the visibility of the objects inside the rack and facilitate their selection and manipulation. In this way, the participants were able to reduce physical movement required to perform the task. This demonstrates how *Move'n'Hold* addresses a fundamental issue in VR where the physical space in which a user can move is often smaller than the movement required to explore the virtual environment.

During the experiments, one participant mentioned that it initially felt strange to move the head to select an object, while another participant explicitly pointed out the comfort of head-movement-based selections (compared to conventional VR-HMD controllers).

Based on our observations, we conclude that the way selection is implemented here is not ideal if the object being selected is very close (i.e., the object covers a large part of the screen and thus requires large head movements for unselection) or very far away (i.e., it covers only a small part of the screen such that hovering it with the selection point

requires careful and precise head movements).

Furthermore, we observed that during detailed planning with the VR-HMD application participants toggled more often between the translation/rotation mode than with the MR-HHD application in the conceptual planning phase.

Again, collisions were rather avoided. Occasionally, an object collided with another object at its back which was not immediately visible and confused the user as the object could not be moved further. When the tablet's movement was slightly directed downwards, some participants also encountered unintended collisions of the object being manipulated with the floor. Thus, preventing object manipulation upon collision detection again rather impeded interaction.

Another interesting observation is that some participants simulated the tasks which a worker in this unit would have to accomplish (i.e., picking up the working material in the rack and moving it to the machine) to check if the space is sufficient, identify ergonomic issues, and thus determine the ideal position of the unit's components.

6.3.4.2 Interviews

After the completion of both the conceptual and the detailed planning tasks, semi-structured interviews were conducted regarding the experienced usability, potential extensions or improvements. In the following we summarize the feedback gained along with the questions asked.

How easy was it to learn the interaction paradigm? The participants stated that, overall, they were able to quickly understand the interaction paradigm and pointed out the benefits of *Move'n'Hold*'s minimalist design. Interestingly, one participant mentioned that the absence of a concrete task in the training application made him assume that he understood and would be able to apply the interaction paradigm too early. Other participants noted that while they found the interaction paradigm simple and straightforward to understand, its correct operation (i.e., coordination tablet movement and touch input) required hands-on practice as this manner of interaction with a tablet is new to them. Nevertheless, they expect to get used to it quickly just like with other interaction paradigms they have learned in the past. One participant initially understood that the tablet works similar as a joystick and

thus expected that objects can be translated by tilting the tablet but was able to learn the correct operation of *Move'n'Hold* quickly. In contrast to physically translating the tablet in space, tilting the tablet would impair the visibility of its screen and thus the MR content.

Do you think that it will be easy for you to relearn the interaction technique? After the experiments, some participants expected that they can resume working with *Move'n'Hold* immediately even after a break of two weeks, whereas other participants believed that they would need some time to recall the interaction paradigm albeit not as much as when they learned it for the first time.

Did you experience cross-device benefits? When the participants were asked if they think that the previous usage of Move'n'Hold with the MR-HHD application helped them to use *Move'n'Hold* with the VR-HMD application, most of the participants reported that they did experience cross-device benefits and confirmed that the usage of *Move'n'Hold* with on device indeed helped when using it with another device. They appreciated that they did not have to learn a new interaction paradigm when moving on to another application. Interestingly, some participants stated that the amount of experienced cross-device benefits did not change between training and conceptual FLP with the MR-HHD and they assumed that they could have immediately proceeded with detailed planning using the VR-HMD application after the training phase. Other participants reported that the training application was of greater benefit for the usage of the VR-HMD application as it supports unconstrained object manipulation whereas the MR-HHD application limits translations to the xz-plane.

How do the XR-tools compare to currently used (2D-)tools for FLP? The participants agreed that the option for more realistic 3D visualizations is the strongest advantage offered by XR as it facilitates the perception and evaluation of distances. They stated that especially the VR-HMD application, which allows performing and testing operations inside the virtual environment, has the potential to outperform existing 2D-tools. Overall, the availability of the consistent interaction paradigm *Move'n'Hold* across planning phases was appreciated and the participants noted that they prefer to not have to adjust to unfamiliar systems. Nevertheless, it

was also noted that performing the conceptual planning task (i.e., arranging factory units) with common mouse-based interaction techniques would not require a lot of effort as these types of interaction are an integral part of their daily work routine. Thus, the MR-HHD application for conceptual planning was not deemed as useful as the VR-HMD application for detailed planning. While some participants believed that they could accomplish the conceptual planning tasks faster with a 2D application and a mouse, they also note that with increasing use of XR technologies in both working and everyday life, their preference may change as well. In this regard it was also acknowledged that other interaction modalities (e.g., mouse-based or touch-based interaction) once required some sort of learning process before they were adopted as standard input techniques.

What is the optimal degree of realism for XR scenes? To gain insights on the ideal degree of realism we asked the participants which of the implemented constraints they considered (not) helpful. Regarding collision prevention, the participants stated that it makes sense to clearly define locations where objects can be placed such that the user is not able to place multiple objects in the same spot. In line with the observations made, the participants also noted that in some cases collision prevention was rather counterproductive. For example, they mentioned that during the conceptual planning task with the MR-HHD application, unit rotations could not be performed when they were already placed close to each other. In this context, it was suggested to adjust the implementation in a way which allows objects to pass through each other but ensures that objects can only be released if they are not colliding with another object or to automatically place the object in the next possible spot. Furthermore, the integration of a visual warning appearing upon the collision of two objects was proposed. The implemented rotation constraints which ensure that objects are only rotated around the y-axis were deemed very helpful by our participants. Furthermore, they found the modeling of gravity, which caused certain objects to drop upon release, appropriate and intuitive to use. The participants argued that objects which can only be placed on the ground, such as the racks and machines in our example, should always drop on release whereas objects which can be placed at different heights, such as the working material, should not automatically drop to the ground outside a rack. All in all, the abstraction level used for the visual representations was deemed to provide an appropriate balance between unrealistic 2D tools and a perfect virtual replica of reality. In line with the

design applied in the prototypes, the participants noted that the abstraction level should generally decrease in later planning phases. Furthermore, our assumption that a perfect replication of reality is not desirable was confirmed by the participants. One of them noted that the maxim *the more realistic, the better* may initially appear appropriate. After using the applications for a while, however, the provided abstraction level felt very convenient.

6.3.4.3 Opportunities for Future Research and Extensions

Based on the insights gained through observation and the participants' statements in the interviews the following opportunities for future research and extensions of the presented prototypes are proposed.

User Onboarding. The experiments revealed *Move'n'Hold* to be well-suited for the FLP task but also showed that the participants had to undergo a learning process before they were able to effectively apply the interaction paradigm. It should therefore be investigated how user onboarding methods can be designed to support the user and streamline the learning process. In this context, the integration of gamification elements may be explored. As mentioned by one of our participants, designers of onboarding applications should furthermore consider the integration of specific tasks which need to be completed during onboarding. In this way, the user can be provided with immediate feedback to improve self-assessment.

Customization. The participants stated that they find the double tap gesture more comfortable for toggling between the manipulation modes as it does not require adjusting the position of the hands holding the tablet. However, they still acknowledged that other users may have other preferences and would appreciate options for customization.

Collaboration-support Features. The observations made during the collaborative planning sessions with the MR-HHD emphasize the relevance of options that allow collaborators to precisely reference scene components and directions. To design such a feature, different approaches could be followed. When the applications of all users rely on the same coordinate system (i.e., sharing the same origin in

the physical space) reference points could be integrated in the scene to make it easier for users to explain and understand which direction they are referring to, for example when explaining in which direction they intend to move a component. This approach would be especially helpful in settings in which collaborators face the scene from different viewpoints. On the other hand, if the collaborators' viewpoints are unlikely to change a lot during the session, user-specific coordinate systems could be set up based on each user's position and viewing direction. In this way, directional indications such as *left* or *right* are interpreted correctly by all users.

Selection. In the interviews some participants mentioned that due to the more colorful scene in the VR-HMD application, the highlighting of selected objects did not stand out as in the MR-HHD application. Consequently, there is a need for researching more advanced visualizations of selected items in such settings. On top of that, options for multi-selection should be investigated. In this context, different modalities including speech recognition and further touch-gestures were proposed by our participants. One participant raised the concern that this may reduce the simplicity of *Move'n'Hold*. When designing and evaluating such additional functions, particular attention should therefore be paid to their effects on the user's cognitive load.

Collision Prevention. Collision prevention as implemented in the presented prototypes turned out to impede interaction in some cases. During the interviews, two alternative approaches were proposed which should be evaluated further. If a user tries to release a selected object while it is colliding with another component, the selected object should either be placed in the closest empty spot or it should keep following the tablet's movement (even when no touch-input is registered) and then be placed automatically as soon as no more collisions with other objects are detected. Furthermore, our participants suggested to define fixed positions and orientations for objects such that upon release objects would snap into the next possible position/orientation. The awareness of collisions, especially at the backside of an object, could be enhanced by the integration of visual warnings.

Design of the VR-HMD app. Some participants suggested to extend the VR-HMD application with feedback mechanisms, similar to the transportation intensity value in the MR-HHD application. A future version of the VR-HMD application could

for example display visual warnings if certain ergonomic standards or minimal distances are not respected in the current factory design. One participant suggested to add walls bordering the factory and to use text labels instead of colors for modeling the relevance of the working material.

Visual Cues during Interaction. One participant suggested to integrate an additional visual cue which displays a preview of the automated continuous manipulation offered by *Move'n'Hold* to help the user in estimating the direction and speed of the movement. On top of that, it was proposed to highlight the position at which an object would be placed if it would be released. To further support the user in manipulating components while monitoring optimization criteria, the respective criteria could be displayed closer to the user's current focus point. For example, the transportation intensity value could be displayed next to the currently manipulated unit. In this context, the ideal balance between the usefulness of information provided through the cues and visual information overload needs to be examined.

Navigation and Zoom. In the presented prototypes *Move'n'Hold* was applied for translating and rotating virtual objects. In the future, *Move'n'Hold* could also be applied to manipulate the user's own position and orientation in the scene (instead of those of virtual objects) to enable scene navigation or zooming. In the VR-HMD application this extension would allow the user to move inside the scene even when the physical space the user is located in is limited. In the MR-HHD application presented here, the same extension could be used for zooming. As such, the user could navigate inside a specific unit to review it in 3D. In this way specific adjustments or constraints could already be set in the conceptual planning phase.

Saving Edits. Future versions of both applications would further benefit from options for saving the edits made. If the current solution is below a defined threshold of the specific optimization criteria, automated procedures could evoke auto-save or encourage the user to save the current version. Furthermore, a feature which can be found in photo-editing software which allows its user to toggle between the edited and the original photo could be useful in the FLP applications as well. Here, the user could toggle between the current scene and the last saved version.

6.4 Discussion

The two practical examples presented in this chapter, demonstrate the applicability of the developed UIs and concepts of XR^S.

The XR^S framework as presented in Chapter 3 proposes the integration of a robotic system to enable off-site users to remotely manipulate physical objects on site. At the same time, it can support on-site users in manipulating large, heavy, or hazardous objects to reduce safety issues. The UI for robot control presented in Chapter 6.2 provides the basis for this feature.

The comparison of the MR-HHD-UI to a Gamepad-UI and a Desktop-UI showed that the MR-HHD-UI required on average less cognitive and temporal effort than the other two UIs. These saved efforts are relevant for increasing productivity and reducing failure in complex real-world tasks which are likely to exceed the operator's cognitive and temporal capacities. Due to its particularly high success rate and its ranking as the preferred UI, the MR-HHD-UI is considered a powerful tool which successfully combines human and robotic capabilities. The operator manages task planning while the robotic system handles repetitive and complex tasks such as optimal path computation and the execution of the robot's movements.

Apart from this, the MR-HHD-UI can also be useful for introducing novices to robotics as they can send high-level commands to the robot and then observe and learn how the robot executes them.

In the presented study, the pick and place tasks were only performed with a MR-HHD-UI and limited to small translations due to the robot's limited reachability. However, the presented MR-HHD-UI can be easily extended to large manipulations including both translations and rotations using *Move'n'Hold* as presented in Chapter 5.

Since *Move'n'Hold* is also available for HMDs, the robot control UI can be further extended to all the main access points (i.e., MR-HHDs, MR-HMDs, and VR-HMDs) of the XR^S framework, allowing its users to manipulate dynamic real components of the scene. Off-site users can then manipulate virtual replicas of real scene components with *Move'n'Hold* in order to command the robotic system to manipulate its physical counterpart accordingly. Furthermore, on-site users who intend to manipulate large, heavy, or hazardous real components can command the robotic system to do so by manipulating virtual overlays similar as presented above.

In the second part of this chapter, concepts of XR^S are applied across FLP use cases and XR technologies. The results of our comprehensive analysis outline how different use cases within a FLP process can benefit from the variety of XR technologies, the core idea behind XR^S.

To cover a broad spectrum of XR technologies and FLP use cases, two prototypes were developed for multi-user conceptual planning with a MR-HHD application and detailed planning with a VR-HMD application. Both applications implement the consistent spatial interaction method *Move'n'Hold* presented in Chapter 5.

Concerning the XR^S framework, this provides an example for the manipulation of dynamic virtual components using the MR-HHD and VR-HMD access points. In the future, the applications may be extended to the manipulation of virtual replicas of real components.

The pilot study conducted demonstrated *Move'n'Hold*'s learnability and applicability and once again confirmed its scalability as the participants were able to intuitively apply the interaction paradigm across different use cases and XR technologies.

The participants of our study named the 3D visualization of a factory as the greatest advantage of XR-supported FLP over conventional planning tools. While the VR-HMD application for detailed planning was therefore deemed more helpful than the MR-HHD application, this also underlines the potential of a feature which allows users to zoom into a unit using the MR-HHD application.

Considering the efforts related to the creation of virtual replicas, another interesting finding concerns the participants' statement that a perfect replication of reality is not desirable. The design decisions made during prototype development such as for example the automated placement of objects upon release, constraining manipulation to certain axes, and the different levels of visual abstraction were found to be appropriate. It also turned out that preventing manipulation upon the collision of two objects (such as in reality) is not always useful. Here, a perfect replication of reality would thus impede interaction.

Instead, replicating certain aspects in an unrealistic manner can actually help leveraging the opportunities offered by digital environments such as easy movements of large objects, temporary storage of objects in any position, or allowing objects to pass through each other for faster manipulations. In the future, the presented applications can be further extended with the proposed navigation and zoom feature, also based on *Move'n'Hold*.

The relevance of these outcomes applies beyond the presented prototypes. As shown in Fig. 6.9, the use cases for which the prototypes were developed share characteristics and requirements with other use cases of the FLP process making the developed features applicable to them as well. On top of that, FLP is a relevant task in many different domains which further broadens the relevance of the analysis, prototype features, and findings from the pilot study.

Our work can also be transferred to industrial use cases other than FLP which involve the creation of physical items (i.e., one of the abstract use cases listed in Fig. 3.3). For instance, in the automotive or aerospace industry, the review of product designs can start in VR and then transition to MR in line with the evolution of the product. For small products, HHDs offer a cheap and portable solution whereas for large, complex products HMDs may be preferrable. Again, the ideal degree of virtuality in collaborative settings may be adapted as discussed for FLP in Chapter 6.3.1. In this way, the results apply to use cases which involve spatial interaction such as prototyping, design reviews, construction planning, or training maintenance and repair tasks across various industries.

Parts of this chapter have been previously published in:

V. M. Memmesheimer, I. T. Chuang, B. Ravani, and A. Ebert (2024). Mixed Reality Handheld Displays for Robot Control: A Comparative Study. In I. L. Nunes (Eds.) Human Factors and Systems Interaction. AHFE (2024) International Conference. AHFE Open Access, vol. 154. AHFE International, USA. doi: 10.54941/ahfe1005380.

V. M. Memmesheimer, M. Klar, H. Subbaraj, B. Ravani, J. C. Aurich, and A. Ebert (2025): Applying Consistent Spatial Interaction Techniques to Factory Layout Planning. In J. Y. C. Chen and G. Fragomeni (Eds.) Virtual, Augmented and Mixed Reality. HCII 2025. Lecture Notes in Computer Science, vol. 15788, pp. 93-112. Springer, Cham. doi: 10.1007/978-3-031-93700-2_7.

Chapter 7

Conclusions

XR encompasses a wide spectrum of technologies including different degrees of virtuality such as MR and VR as well as different devices like HMDs and HHDs. Despite the wide range of applications which this technological variety opens up, XR's adoption in practical settings is still limited.

While an appropriate UI design alone does not guarantee the adoption of a technology, it can be a major contributing factor. This makes the creation of easy (i.e., effortless) to use UIs a highly relevant research topic. In the past, considerable research has been conducted to enhance the usability of UIs for specific XR technologies, whereas comparably less research has focused on the development of scalable UIs which enable effortless switching between different XR technologies. Yet, XR's intrinsic potential lies in its variety of technologies which offer distinct benefits for different use cases. Equipping users with the most suitable technology for a specific use case therefore makes seamless switching between XR technologies a key requirement.

Unlike previous research, the design of the scalable UIs proposed in this dissertation considers both the efforts occurring while a particular UI is used as well as those efforts which are required to switch between different UIs. This leads to the following key contributions.

First, Chapter 3 introduces XR^S as a novel concept for XR environments which provide scalability across different degrees of virtuality, different devices, and different numbers of potentially distributed users. Along with the concept of XR^S, we summarize challenges to the realization of XR^S and transfer them into a research agenda. Addressing the first items on the agenda, a use case analysis was carried out to formulate requirements. These were translated into an XR^S framework which serves as the basis for this dissertation. In the

following chapters, essential parts of the framework are realized while addressing further items from the research agenda.

Chapter 4 is focused on collaboration support features which provide consistent awareness cues across the access points of XR^S spaces and ensure that they accurately represent user behavior while avoiding visual overload in large groups. Here, a major challenge concerns the accurate representation of HHD users. We therefore explored how data obtained from an HHD can be transferred into correct representations of an HHD user's activities. To this end, a detailed study was conducted which delivered new insights on how users interact with HHDs across different display sizes, display orientations, and body poses. Our investigations show positive results on the front camera's accessibility during interaction, indicating the potential to integrate face tracking for enhanced user representations. Furthermore, the results show that a ray originating from the device center is, in general, a good proxy for the user's viewing direction but its accuracy is systematically affected by device orientation, viewing direction, and distance. Based on these findings, we derived guidelines for enhanced HHD user representations. The second part of the chapter is focused on the avoidance of visual overload in large groups of collaborators. To this end, we designed mechanisms which adapt the visibility of awareness cues such as head-rays and hand-rays based on natural cooperation paradigms. Thereby, cues of collaborators who a facing the same part of the scene are enabled automatically. At the same time, users can actively enable cues for a specific collaborator and draw attention to themselves. As such, the amount of awareness cues describing the behavior of other collaborators is configured individually for each user. Taking into account the insights gained on HHD user behavior and the developed mechanisms, we discuss considerations for implementing cue visibility handling in settings which involve different XR technologies and collaboration styles.

In Chapter 5 we present the design, implementation, and evaluation of the novel interaction paradigm *Move'n'Hold* which tackles the lack of consistent object manipulation techniques and allows users to seamlessly switch between the access points of XR^S. *Move'n'Hold* was first developed for MR-HHDs where it solves key issues of existing manipulation methods. Our method allows users to hold the device with both hands while performing object translations or rotations with a unified interaction paradigm which is solely based on peripheral touch input and device movements. Despite its minimalist design *Move'n'Hold* offers a huge variety of interactions including seamless combinations of direct and continuous object manipulation. As such it enables users to handle tasks that vary in distance, directions, and complexities. We then extended HMDs with a tablet con-

troller which implements the same interaction paradigm as *Move'n'Hold* for MR-HHDs. This yields consistent interaction across MR-HHDs, MR-HMDs, and VR-HMDs. In a detailed study, comparing the set of *Move'n'Hold* interaction techniques to a corresponding set of state-of-the-art techniques, *Move'n'Hold* was the preferred interaction modality and rated to be easier to relearn. Furthermore, it reduced the workload, improved usability, and provided more cross-device benefits which facilitate switching between devices and degrees of virtuality. Initial investigations have also indicated the potential use of *Move'n'Hold* for scene navigation in the future (i.e., combining touch input and device movements to adapt the user's position and orientation in the scene instead of those of an object).

Eventually, Chapter 6 showcases the applicability of the developed scalable UIs in two practical examples. First, we present how robot control in a pick and place task can be enabled through a MR-HHD-UI. Our comparative study revealed the proposed UI to be a powerful tool which successfully combines human and robotic capabilities. Based on these results, it is outlined how the developed UIs can be integrated for the robotic system proposed in the XR^S framework. In a second application, concepts of XR^S are applied to factory layout planning (FLP). To this end, we first performed a comprehensive analysis to derive recommendations for choosing XR technologies in different use cases. To demonstrate how users can switch between XR technologies and use cases, we then implemented the scalable interaction paradigm *Move'n'Hold* in a multi-user MR-HHD application for conceptual FLP and in a VR-HMD application for detailed FLP. The conducted pilot study again confirmed *Move'n'Hold*'s scalability as it showed that participants could intuitively apply the interaction paradigm across different XR technologies and use cases. Eventually, we provide design recommendations as well as opportunities for future extensions and research based on the observations made and the interviews conducted.

Beyond the provided examples, the contributions of this dissertation can find application in numerous other domains. Possible use cases can be described by (combinations of) the abstract use cases on which the XR^S framework was built. As such, promising fields of application include but are not limited to the aerospace, automotive, chemical, pharmaceutical, or food industry, construction and architecture, as well as the medical domain. Facility design and construction (similar as presented for FLP) in Chapter 6.3 is not only relevant in different manufacturing industries but also in public and private domains. Similar UIs have strong potential in the (collaborative) creation and review of product designs. Thereby, large products such as the interior and exterior design of

vehicles, aircrafts, or spacecrafts can be visualized and reviewed at full scale whereas small products such as wearables, machine parts, or medical devices can be displayed in enlarged form. Additional applications can be found in training scenarios, for example, those involving maintenance, repair, or assembly tasks in different manufacturing industries as well as the teleoperation of machines, robotic systems, and drones in inaccessible or hazardous environments. The integration of digital twins and simulations into the XR environment opens additional opportunities to enhance virtual replicas and available information. Finally, further interesting (collaborative) use cases arise in the context of immersive analytics, crime scene investigation, surgical training and planning, as well as training and teleoperation for space exploration. Ultimately, this variety of potential applications for the developed scalable UIs implies a correspondingly broad user base.

As outlined in Chapter 3, the realization of XR^S involves research and innovations in several fields. The future adoption of XR technologies will also depend on hardware manufacturers addressing key requirements such as accessibility, ergonomics, display size and resolution, and developer support. Apart from enhanced hardware, remaining items from the research agenda can serve as starting points for future work – for example, enhancing virtual replicas, designing and evaluating further collaboration support features, or expanding research on user onboarding.

Bibliography

[1] K. Ahuja, M. Goel, and C. Harrison. BodySLAM: Opportunistic user digitization in multi-user AR/VR experiences. In *Proceedings of the 2020 ACM Symposium on Spatial User Interaction*, SUI '20, pages 1–8, Article 16, New York, NY, USA, 2020. ACM. doi: 10.1145/3385959.3418452.

[2] K. Ahuja, S. Mayer, M. Goel, and C. Harrison. Pose-on-the-Go: Approximating user pose with smartphone sensor fusion and inverse kinematics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–12, Article 9, New York, NY, USA, 2021. ACM. doi: 10.1145/3411764.3445582.

[3] H. F. Al Janabi, A. Aydin, S. Palaneer, N. Macchione, A. Al-Jabir, M. S. Khan, P. Dasgupta, and K. Ahmed. Effectiveness of the hololens mixed-reality headset in minimally invasive surgery: a simulation-based feasibility study. *Surgical Endoscopy*, 34:1143–1149, 2020. doi: 10.1007/s00464-019-06862-3.

[4] T. Babic, F. Perteneder, H. Reiterer, and M. Haller. Simo: Interactions with distant displays by smartphones with simultaneous face and world tracking. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pages 1–12, New York, NY, USA, 2020. ACM. doi: 10.1145/3334480.3382962.

[5] H. Bai, P. Sasikumar, J. Yang, and M. Billinghurst. A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376550.

[6] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, 2008. doi: 10.1080/10447310802205776.

[7] D. K. Baroroh and C.-H. Chu. Human-centric production system simulation in

mixed reality: An exemplary case of logistic facility design. *Journal of Manufacturing Systems*, 65:146–157, 2022. doi: 10.1016/j.jmsy.2022.09.005.

[8] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 481–488, New York, NY, USA, 2003. ACM. doi: 10.1145/642611.642695.

[9] F. Bellalouna. New approach for digital factory using virtual reality technology. *Procedia CIRP*, 93:256–261, 2020. doi: 10.1016/j.procir.2020.04.012.

[10] S. Benford, J. Bowers, L. E. Fahlén, C. Greenhalgh, and D. Snowdon. User embodiment in collaborative virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 242–249, USA, 1995. ACM Press/Addison-Wesley Publishing Co. doi: 10.1145/223904.223935.

[11] L. Besançon, M. Sereno, L. Yu, M. Ammi, and T. Isenberg. Hybrid touch/tangible spatial 3D data selection. *Computer Graphics Forum*, 38(3):553–567, 2019. doi: 10.1111/cgf.13710.

[12] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3490–3502, 2020. doi: 10.1109/TVCG.2020.3023567.

[13] O. Bimber and R. Raskar. *Spatial Augmented Reality - Merging Real and Virtual Worlds*. A K Peters, 2005.

[14] J. Blattgerste, P. Renner, and T. Pfeiffer. 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*, COGAIN '18, pages 1–9, Article 1, New York, NY, USA, 2018. ACM. doi: 10.1145/3206343.3206349.

[15] J. Blattgerste, K. Luksch, C. Lewa, and T. Pfeiffer. TrainAR: A scalable interaction concept and didactic framework for procedural trainings using handheld augmented reality. *Multimodal Technologies and Interaction*, 5(7):30, 2021. doi: 10.3390/mti5070030.

[16] J. Botev, J. Mayer, and S. Rothkugel. Immersive mixed reality object interaction for collaborative context-aware mobile training and exploration. In *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems*, MMVE '19, pages 4–9, New York, NY, USA, 2019. ACM. doi: 10.1145/3304113.3326117.

[17] R. Bovo, D. Giunchi, M. Alebri, A. Steed, E. Costanza, and T. Heinis. Cone of vision as a behavioural cue for VR collaboration. *Proc. ACM Hum.-Comput. Interact.*, 6 (CSCW2):1–27, Article 502, 2022. doi: 10.1145/3555615.

[18] D. A. Bowman, E. Kruijff, J. J. LaViola Jr., and I. Poupyrev. *3D User Interfaces - Theory and Practice*. Pearson Education, Inc., 2005.

[19] E. Bozgeyikli and L. L. Bozgeyikli. Evaluating object manipulation interaction techniques in mixed reality: Tangible user interfaces and gesture. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 778–787. IEEE, 2021. doi: 10.1109/VR50410.2021.00105.

[20] E. Brasier, O. Chapuis, N. Ferey, J. Vezien, and C. Appert. ARPads: Mid-air indirect input for augmented reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 332–343. IEEE, 2020. doi: 10.1109/ISMAR50242.2020.00060.

[21] J. Brooke. SUS: A quick and dirty usability scale. In *Usability Evaluation In Industry*, pages 189–194. Taylor and Francis, London, 1996.

[22] P. Burggräf and G. Schuh (Eds.). *Fabrikplanung - Handbuch Produktion und Management 4, 2. Auflage*. Springer Vieweg, 2021. doi: 10.1007/978-3-662-61969-8.

[23] M. Cai and J. Tanaka. Go together: providing nonverbal awareness cues to enhance co-located sensation in remote communication. *Human-centric Computing and Information Sciences*, 9:19, 2019. doi: 10.1186/s13673-019-0180-y.

[24] Y. Cao, Z. Xu, F. Li, W. Zhong, K. Huo, and K. Ramani. V.Ra: An in-situ visual authoring system for Robot-IoT task planning with augmented reality. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, pages 1059–1070, New York, NY, USA, 2019. ACM. doi: 10.1145/3322276.3322278.

[25] S. M. Chacko and V. Kapila. An augmented reality interface for human-robot interaction in unconstrained environments. In *2019 IEEE/RSJ International Con-*

ference on Intelligent Robots and Systems (IROS), pages 3222–3228. IEEE, 2019. doi: 10.1109/IROS40897.2019.8967973.

[26] N. Chaconas and T. Höllerer. An evaluation of bimanual gestures on the microsoft hololens. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8. IEEE, 2018. doi: 10.1109/VR.2018.8446320.

[27] W. P. Chan, G. Hanks, M. Sakr, T. Zuo, H. F. Machiel Van der Loos, and E. Croft. An augmented reality human-robot physical collaboration interface design for shared, large-scale, labour-intensive manufacturing tasks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11308–11313. IEEE, 2020. doi: 10.1109/IROS45743.2020.9341119.

[28] S. Chandra Sekaran, H. J. Yap, S. N. Musa, K. E. Liew, C. H. Tan, and A. Aman. The implementation of virtual reality in digital factory—a comprehensive review. *The International Journal of Advanced Manufacturing Technology*, 115:1349–1366, 2021. doi: 10.1007/s00170-021-07240-x.

[29] L. Chen, K. Takashima, K. Fujita, and Y. Kitamura. PinpointFly: An egocentric position-control drone interface using mobile AR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–13, Article 150, New York, NY, USA, 2021. ACM. doi: 10.1145/3411764.3445110.

[30] Y. Chen, K. Katsuragawa, and E. Lank. Understanding viewport- and world-based pointing with everyday smart devices in immersive augmented reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376592.

[31] M. Cordeil, B. Bach, A. Cunningham, B. Montoya, R. T. Smith, B. H. Thomas, and T. Dwyer. Embodied axes: Tangible, actuated interaction for 3D augmented reality data spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376613.

[32] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 135–142, New York, NY, USA, 1993. ACM. doi: 10.1145/166117.166134.

[33] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443. IEEE, 2017. doi: 10.1109/CVPR.2017.261.

[34] F. D. Davis. *A technology acceptance model for empirically testing new end-user information systems: theory and results*. PhD thesis, MIT, Cambridge, MA, USA, 1986.

[35] F. D. Davis. On the relationship between HCI and technology acceptance research. In *Zhang, P., Galletta, D. (Eds.) Human-Computer Interaction and Management Information Systems: Foundations. Advances in Management Information Systems*, volume 5, pages 395–401. M.E. Sharpe, Armonk, NY, USA, 2006.

[36] A. S. L. de Andrade, V. Jackson, R. Prikladnicki, and A. van der Hoek. On meetings involving remote software teams: A systematic literature review. *Information and Software Technology*, 175:107541, 2024. doi: 10.1016/j.infsof.2024.107541.

[37] F. De Pace, F. Manuri, A. Sanna, and D. Zappia. A comparison between two different approaches for a collaborative mixed-virtual environment in industrial maintenance. *Frontiers in Robotics and AI*, 6:18, 2019. doi: 10.3389/frobt.2019.00018.

[38] E. DeFilippis, S. M. Impink, M. Singell, J. T. Polzer, and R. Sadun. The impact of COVID-19 on digital communication patterns. *Humanities and Social Sciences Communications*, 9:180, 2022. doi: 10.1057/s41599-022-01190-9.

[39] DIN. Ergonomie der Mensch-System-Interaktion – Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte (ISO 9241-11:2018); Deutsche Fassung EN ISO 9241-11:2018, 2018.

[40] DIN. Ergonomie der Mensch-System-Interaktion – Teil 110: Interaktionsprinzipien (ISO 9241-110:2020); Deutsche Fassung EN ISO 9241-110:2020, 2020.

[41] M. T. Dishaw and D. M. Strong. Extending the technology acceptance model with task–technology fit constructs. *Information & Management*, 36(1):9–21, 1999. doi: 10.1016/S0378-7206(98)00101-3.

[42] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction, Third Edition*. Pearson Education Limited, 2004.

[43] M. Dixken, D. Diers, B. Wingert, A. Hatzipanayioti, B. J. Mohler, O. Riedel, and M. Bues. Distributed, collaborative virtual reality application for product development with simple avatar calibration method. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1299–1300. IEEE, 2019. doi: 10.1109/VR.2019.8797884.

[44] T. Drey, P. Albus, S. der Kinderen, M. Milo, T. Segschneider, L. Chanzab, M. Rietzler, T. Seufert, and E. Rukzio. Towards collaborative learning in virtual reality: A comparison of co-located symmetric and asymmetric pair-learning. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–19, Article 610, New York, NY, USA, 2022. ACM. doi: 10.1145/3491102.3517641.

[45] R. Dörner, W. Broll, P. Grimm, and B. Jung (Eds.). *Virtual und Augmented Reality (VR/AR) - Grundlagen und Methoden der Virtuellen und Augmentierten Realität, 2. Auflage*. Springer Vieweg, 2019. doi: 10.1007/978-3-662-58861-1.

[46] M. Emporio, A. Caputo, D. Pintani, D. S. Cheng, T. De Marchi, G. Forte, F. Fummi, and A. Giachetti. Integration of extended reality with a cyber-physical factory environment and its digital twins. *Proc. ACM Hum.-Comput. Interact.*, 8(EICS): 1–13, Article 250, 2024. doi: 10.1145/3660246.

[47] D. Englmeier, J. Dörner, A. Butz, and T. Höllerer. A tangible spherical proxy for object manipulation in augmented reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 221–229. IEEE, 2020. doi: 10.1109/VR46266.2020.00041.

[48] A. Erickson, N. Norouzi, K. Kim, R. Schubert, J. Jules, J. J. LaViola, G. Bruder, and G. F. Welch. Sharing gaze rays for visual target identification tasks in collaborative augmented reality. *Journal on Multimodal User Interfaces*, 14:353–371, 2020. doi: 10.1007/s12193-020-00330-2.

[49] A. Esteves, Y. Shin, and I. Oakley. Comparing selection mechanisms for gaze input techniques in head-mounted displays. *International Journal of Human-Computer Studies*, 139:102414, 2020. doi: 10.1016/j.ijhcs.2020.102414.

[50] M. Eswaran and M. V. A. Raju Bahubalendruni. Challenges and opportunities on AR/VR technologies for manufacturing systems in the context of industry 4.0: A state of the art review. *Journal of Manufacturing Systems*, 65:260–278, 2022. doi: 10.1016/j.jmsy.2022.09.016.

[51] J. A. Frank, S. P. Krishnamoorthy, and V. Kapila. Toward mobile mixed-reality interaction with multi-robot systems. *IEEE Robotics and Automation Letters*, 2(4): 1901–1908, 2017. doi: 10.1109/LRA.2017.2714128.

[52] J. A. Frank, M. Moorhead, and V. Kapila. Mobile mixed-reality interfaces that enhance human–robot interaction in shared spaces. *Frontiers in Robotics and AI*, 4:20, 2017. doi: 10.3389/frobt.2017.00020.

[53] Free3D. URL `https://free3d.com`.

[54] A. Fuste, B. Reynolds, J. Hobin, and V. Heun. Kinetic AR: A framework for robotic motion systems in spatial computing. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pages 1–8, New York, NY, USA, 2020. ACM. doi: 10.1145/3334480.3382814.

[55] V. Fuvattanasilp, Y. Fujimoto, A. Plopski, T. Taketomi, C. Sandor, M. Kanbara, and H. Kato. SlidAR+: Gravity-aware 3D object manipulation for handheld augmented reality. *Computers & Graphics*, 95:23–35, 2021. doi: 10.1016/j.cag.2021.01.005.

[56] L. Gao, H. Bai, W. He, M. Billinghurst, and R. W. Lindeman. Real-time visual representations for mobile mixed reality remote collaboration. In *SIGGRAPH Asia 2018 Virtual & Augmented Reality*, SA '18, pages 1–2, Article 15, New York, NY, USA, 2018. ACM. doi: 10.1145/3275495.3275515.

[57] I. García-Pereira, J. Gimeno, M. Pérez, C. Portalés, and S. Casas. MIME: A mixed-space collaborative system with three immersion levels and multiple users. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 179–183. IEEE, 2018. doi: 10.1109/ISMAR-Adjunct.2018.00062.

[58] S. Gauglitz, C. Lee, M. Turk, and T. Höllerer. Integrating the physical environment into mobile remote collaboration. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 241–250, New York, NY, USA, 2012. ACM. doi: 10.1145/2371574.2371610.

[59] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer. In touch with the remote world: remote collaboration with augmented reality drawings and virtual navigation. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and*

Technology, VRST '14, pages 197–205, New York, NY, USA, 2014. ACM. doi: 10.1145/2671015.2671016.

[60] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer. World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 449–459, New York, NY, USA, 2014. ACM. doi: 10.1145/2642918.2647372.

[61] C. George, A. N. Tien, and H. Hussmann. Seamless, bi-directional transitions along the reality-virtuality continuum: A conceptualization and prototype exploration. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 412–424. IEEE, 2020. doi: 10.1109/ISMAR50242.2020.00067.

[62] E. S. Goh, M. S. Sunar, and A. W. Ismail. 3D object manipulation techniques in handheld mobile augmented reality interface: A review. *IEEE Access*, 7:40581–40601, 2019. doi: 10.1109/ACCESS.2019.2906394.

[63] E. B. Goldstein. *Cognitive Psychology, Third Edition*. Wadsworth, Cengage Learning, 2011.

[64] L. Gong, J. Berglund, Å. Fast-Berglund, B. Johansson, Z. Wang, and T. Börjesson. Development of virtual reality support to factory layout planning. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13:935–945, 2019. doi: 10.1007/s12008-019-00538-x.

[65] L. Gong, H. Söderlund, L. Bogojevic, X. Chen, A. Berce, Å. Fast-Berglund, and B. Johansson. Interaction design for multi-user virtual reality systems: An automotive case study. *Procedia CIRP*, 93:1259–1264, 2020. doi: 10.1016/j.procir.2020.04.036.

[66] D. L. Goodhue and R. L. Thompson. Task-technology fit and individual performance. *MIS Quarterly*, 19(2):213–236, 1995. doi: 10.2307/249689.

[67] J. G. Grandi, H. G. Debarba, L. Nedel, and A. Maciel. Design and evaluation of a handheld-based 3D user interface for collaborative object manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5881–5891, New York, NY, USA, 2017. ACM. doi: 10.1145/3025453.3025935.

[68] J. G. Grandi, H. G. Debarba, I. Bemdt, L. Nedel, and A. Maciel. Design and assessment of a collaborative 3D interaction technique for handheld augmented reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 49–56. IEEE, 2018. doi: 10.1109/VR.2018.8446295.

[69] R. Grasset, J. Looser, and M. Billinghurst. Transitional interface: concept, issues and framework. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 231–232. IEEE, 2006. doi: 10.1109/ISMAR.2006.297819.

[70] R. A. Grier. How high is high? A meta-analysis of NASA-TLX global workload scores. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59(1):1727–1731, 2015. doi: 10.1177/1541931215591373.

[71] S. Günther, S. Kratz, D. Avrahami, and M. Mühlhäuser. Exploring audio, visual, and tactile cues for synchronous remote assistance. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, PETRA '18, pages 339–344, New York, NY, USA, 2018. ACM. doi: 10.1145/3197768.3201568.

[72] D. Herr, J. Reinhardt, G. Reina, R. Krüger, R. V. Ferrari, and T. Ertl. Immersive modular factory layout planning using augmented reality. *Procedia CIRP*, 72:1112–1117, 2018. doi: 10.1016/j.procir.2018.03.200.

[73] J. Hertel, S. Karaosmanoglu, S. Schmidt, J. Bräker, M. Semmann, and F. Steinicke. A taxonomy of interaction techniques for immersive augmented reality based on an iterative literature review. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 431–440. IEEE, 2021. doi: 10.1109/ISMAR52148.2021.00060.

[74] J. Hindmarsh, M. Fraser, C. Heath, S. Benford, and C. Greenhalgh. Fragmented interaction: establishing mutual orientation in virtual environments. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 217–226, New York, NY, USA, 1998. ACM. doi: 10.1145/289444.289496.

[75] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu. Supervoxel convolution for online 3D semantic segmentation. *ACM Trans. Graph.*, 40(3):1–15, Article 34, 2021. doi: 10.1145/3453485.

[76] W. Huang, M. Wakefield, T. A. Rasmussen, S. Kim, and M. Billinghurst. A review on communication cues for augmented reality based remote guidance. *Journal on Multimodal User Interfaces*, 16:239–256, 2022. doi: 10.1007/s12193-022-00387-1.

[77] S. Hubenschmid, J. Zagermann, S. Butscher, and H. Reiterer. STREAM: Exploring the combination of spatially-aware tablets with augmented reality head-mounted displays for immersive analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–14, Article 469, New York, NY, USA, 2021. ACM. doi: 10.1145/3411764.3445298.

[78] S. Hueber, C. Cherek, P. Wacker, J. Borchers, and S. Voelker. Headbang: Using head gestures to trigger discrete actions on mobile devices. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '20, pages 1–10, Article 17, New York, NY, USA, 2020. ACM. doi: 10.1145/3379503.3403538.

[79] H. Ibayashi, Y. Sugiura, D. Sakamoto, N. Miyata, M. Tada, T. Okuma, T. Kurata, M. Mochimaru, and T. Igarashi. Dollhouse VR: a multi-view, multi-user collaborative design workspace with VR technology. In *SIGGRAPH Asia 2015 Emerging Technologies*, SA '15, pages 1–2, Article 8, New York, NY, USA, 2015. ACM. doi: 10.1145/2818466.2818480.

[80] R. Jacob and S. Stellmach. What you look at is what you get: gaze-based user interfaces. *Interactions*, 23(5):62–65, 2016. doi: 10.1145/2978577.

[81] R. J. K. Jacob. 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*, CHI '90, pages 11–18, New York, NY, USA, 1990. ACM. doi: 10.1145/97243.97246.

[82] S. Jeršov and A. Tepljakov. Digital twins in extended reality for control system applications. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 274–279. IEEE, 2020. doi: 10.1109/TSP49548.2020.9163557.

[83] H.-C. Jetter, J.-H. Schröder, J. Gugenheimer, M. Billinghurst, C. Anthes, M. Khamis, and T. Feuchtner. Transitional interfaces in mixed and cross-reality: A new frontier? In *Companion Proceedings of the 2021 Conference on Interactive Surfaces and Spaces*, ISS Companion '21, pages 46–49, New York, NY, USA, 2021. ACM. doi: 10.1145/3447932.3487940.

[84] A. Jing, K. May, B. Matthews, G. Lee, and M. Billinghurst. The impact of sharing gaze behaviours in collaborative mixed reality. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2):1–27, Article 463, 2022. doi: 10.1145/3555564.

[85] R. Johansen. Teams for tomorrow (groupware). In *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, volume 3, pages 521–534. IEEE, 1991. doi: 10.1109/HICSS.1991.184183.

[86] A. Kaluza, M. Juraschek, L. Büth, F. Cerdas, and C. Herrmann. Implementing mixed reality in automotive life cycle engineering: A visual analytics based approach. *Procedia CIRP*, 80:717–722, 2019. doi: 10.1016/j.procir.2019.01.078.

[87] H. J. Kang, J.-h. Shin, and K. Ponto. A comparative analysis of 3D user interaction: How to move virtual objects in mixed reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 275–284. IEEE, 2020. doi: 10.1109/VR46266.2020.00047.

[88] M. Kapinus, V. Beran, Z. Materna, and D. Bambušek. Spatially situated end-user robot programming in augmented reality. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–8. IEEE, 2019. doi: 10.1109/RO-MAN46459.2019.8956336.

[89] M. Kari and C. Holz. HandyCast: Phone-based bimanual input for virtual reality in mobile and space-constrained settings via pose-and-touch transfer. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pages 1–15, Article 528, New York, NY, USA, 2023. ACM. doi: 10.1145/3544548.3580677.

[90] M. Kim and J. Y. Lee. Touch and hand gesture-based interactions for directly manipulating 3D virtual objects in mobile augmented reality. *Multimedia Tools and Applications*, 75:16529–16550, 2016. doi: 10.1007/s11042-016-3355-9.

[91] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, and M. Billinghurst. Evaluating the combination of visual communication cues for HMD-based mixed reality remote collaboration. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–13, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300403.

[92] S. Kim, G. Lee, M. Billinghurst, and W. Huang. The combination of visual communication cues in mixed reality remote collaboration. *Journal on Multimodal User Interfaces*, 14:321–335, 2020. doi: 10.1007/s12193-020-00335-x.

[93] F. Kiss, P. W. Woźniak, V. Biener, P. Knierim, and A. Schmidt. VUM: Understanding requirements for a virtual ubiquitous microscope. In *Proceedings of the*

19th International Conference on Mobile and Ubiquitous Multimedia, MUM '20, pages 259–266, New York, NY, USA, 2020. ACM. doi: 10.1145/3428361.3428386.

[94] P. Knierim, D. Hein, A. Schmidt, and T. Kosch. The SmARtphone controller: Leveraging smartphones as input and output modality for improved interaction within mobile augmented reality environments. *i-com*, 20(1):49–61, 2021. doi: 10.1515/icom-2021-0003.

[95] A. Kokkas and G.-C. Vosniakos. An augmented reality approach to factory layout design embedding operation simulation. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13:1061–1071, 2019. doi: 10.1007/s12008-019-00567-6.

[96] G. Kostov and J. Wolfartsberger. Designing a framework for collaborative mixed reality training. *Procedia Computer Science*, 200:896–903, 2022. doi: 10.1016/j.procs.2022.01.287.

[97] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–14, New York, NY, USA, 2018. ACM. doi: 10.1145/3173574.3173655.

[98] K.-D. Le, T. Q. Tran, K. Chlasta, K. Krejtz, M. Fjeld, and A. Kunz. VXSlate: Exploring combination of head movements and mobile touch for large virtual display interaction. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, DIS '21, pages 283–297, New York, NY, USA, 2021. ACM. doi: 10.1145/3461778.3462076.

[99] G. A. Lee, T. Teo, S. Kim, and M. Billinghurst. A user study on MR remote collaboration using live 360 video. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 153–164. IEEE, 2018. doi: 10.1109/ISMAR.2018.00051.

[100] L.-H. Lee, Y. Zhu, Y.-P. Yau, T. Braud, X. Su, and P. Hui. One-thumb text acquisition on force-assisted miniature interfaces for mobile headsets. In *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2020. doi: 10.1109/PerCom45495.2020.9127378.

[101] H.-N. Liang, Y. Shi, F. Lu, J. Yang, and K. Papangelis. VRMController: an input device for navigation activities in virtual reality environments. In *Proceedings of*

the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1, VRCAI '16, pages 455–460, New York, NY, USA, 2016. ACM. doi: 10.1145/3013971.3014005.

[102] K.-C. Lien, B. Nuernberger, T. Höllerer, and M. Turk. PPV: Pixel-point-volume segmentation for object referencing in collaborative augmented reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 77–83. IEEE, 2016. doi: 10.1109/ISMAR.2016.21.

[103] D. Lindlbauer and A. D. Wilson. Remixed reality: Manipulating space and time in augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, New York, NY, USA, 2018. ACM. doi: 10.1145/3173574.3173703.

[104] W. Luo, E. Goebel, P. Reipschläger, M. O. Ellenberg, and R. Dachselt. Exploring and slicing volumetric medical data in augmented reality using a spatially-aware mobile device. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 334–339. IEEE, 2021. doi: 10.1109/ISMAR-Adjunct54149.2021.00076.

[105] E. Makled, F. Weidner, and W. Broll. Investigating user embodiment of inverse-kinematic avatars in smartphone augmented reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 666–675. IEEE, 2022. doi: 10.1109/ISMAR55827.2022.00084.

[106] S. Mann. Mediated reality. *Linux J.*, 1999(59es):5–es, 1999.

[107] S. Marks and D. White. Multi-device collaboration in virtual environments. In *Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*, ICVARS '20, pages 35–38, New York, NY, USA, 2020. ACM. doi: 10.1145/3385378.3385381.

[108] A. Martinet, G. Casiez, and L. Grisoni. The effect of dof separation in 3D manipulation tasks with multi-touch displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pages 111–118, New York, NY, USA, 2010. ACM. doi: 10.1145/1889863.1889888.

[109] A. Marzo, B. Bossavit, and M. Hachet. Combining multi-touch input and device movement for 3D manipulations in mobile augmented reality environments. In

Proceedings of the 2nd ACM Symposium on Spatial User Interaction, SUI '14, pages 13–16, New York, NY, USA, 2014. ACM. doi: 10.1145/2659766.2659775.

[110] F. Matulic, A. Ganeshan, H. Fujiwara, and D. Vogel. Phonetroller: Visual representations of fingers for precise touch input with mobile phones in VR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–13, Article 129, New York, NY, USA, 2021. ACM. doi: 10.1145/3411764.3445583.

[111] S. Mayer, G. Laput, and C. Harrison. Enhancing mobile voice assistants with WorldGaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–10, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376479.

[112] A. Meier, H. Spada, and N. Rummel. A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning*, 2:63–86, 2007. doi: 10.1007/s11412-006-9005-x.

[113] D. Mendes, F. Relvas, A. Ferreira, and J. Jorge. The benefits of DOF separation in mid-air 3D object manipulation. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 261–268, New York, NY, USA, 2016. ACM. doi: 10.1145/2993369.2993396.

[114] D. Mendes, M. Sousa, R. Lorena, A. Ferreira, and J. Jorge. Using custom transformation axes for mid-air manipulation of 3D virtual objects. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, pages 1–8, Article 27, New York, NY, USA, 2017. ACM. doi: 10.1145/3139131.3139157.

[115] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[116] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. In *H. Das (Ed.) Telemanipulator and Telepresence Technologies*, volume 2351, pages 282 – 292. SPIE, 1995. doi: 10.1117/12.197321.

[117] P. Mohan, W. B. Goh, C.-W. Fu, and S.-K. Yeung. DualGaze: Addressing the midas touch problem in gaze mediated VR interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 79–84. IEEE, 2018. doi: 10.1109/ISMAR-Adjunct.2018.00039.

[118] P. Mohr, S. Mori, T. Langlotz, B. H. Thomas, D. Schmalstieg, and D. Kalkofen. Mixed reality light fields for interactive remote assistance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376289.

[119] M. Mori, K. F. MacDorman, and N. Kageki. The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100, 2012. doi: 10.1109/MRA.2012.2192811.

[120] A. Mossel, B. Venditti, and H. Kaufmann. 3DTouch and HOMER-S: Intuitive manipulation techniques for one-handed handheld augmented reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*, VRIC '13, pages 1–10, Article 12, New York, NY, USA, 2013. ACM. doi: 10.1145/2466816.2466829.

[121] F. Müller, J. McManus, S. Günther, M. Schmitz, M. Mühlhäuser, and M. Funk. 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*, CHI '19, pages 1–13, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300707.

[122] A. Murugan, R. Vanukuru, and J. Pillai. Towards avatars for remote communication using mobile augmented reality. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 135–139. IEEE, 2021. doi: 10.1109/VRW52623.2021.00032.

[123] D. Nåfors, B. Johansson, P. Gullander, and S. Erixon. Simulation in hybrid digital twins for factory layout planning. In *2020 Winter Simulation Conference (WSC)*, pages 1619–1630. IEEE, 2020. doi: 10.1109/WSC48552.2020.9384075.

[124] T. Nagai, K. Fujita, K. Takashima, and Y. Kitamura. HandyGaze: A gaze tracking technique for room-scale environments using a single smartphone. *Proc. ACM Hum.-Comput. Interact.*, 6(ISS):1–18, Article 562, 2022. doi: 10.1145/3567715.

[125] NASA. NASA TLX Paper and Pencil Version Instruction Manual, 2025. URL `https://humansystems.arc.nasa.gov/groups/tlx/tlxpaperpencil.php`. accessed June 9, 2025.

[126] A. Naumann and J. Hurtienne. Benchmarks for intuitive interaction with mobile devices. In *Proceedings of the 12th International Conference on Human Computer*

Interaction with Mobile Devices and Services, MobileHCI '10, pages 401–402, New York, NY, USA, 2010. ACM. doi: 10.1145/1851600.1851685.

[127] J. Nielsen. Why you only need to test with 5 users, 2000. URL `https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`. accessed Dec 22, 2024.

[128] B. Nuernberger, K.-C. Lien, L. Grinta, C. Sweeney, M. Turk, and T. Höllerer. Multi-view gesture annotations in image-based 3D reconstructed scenes. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 129–138, New York, NY, USA, 2016. ACM. doi: 10.1145/2993369.2993371.

[129] B. Nuernberger, K.-C. Lien, T. Höllerer, and M. Turk. Interpreting 2D gesture annotations in 3D augmented reality. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 149–158. IEEE, 2016. doi: 10.1109/3DUI.2016.7460046.

[130] T. Nukarinen, J. Kangas, J. Rantala, O. Koskinen, and R. Raisamo. Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, pages 1–2, Article 86, New York, NY, USA, 2018. ACM. doi: 10.1145/3281505.3283382.

[131] A. Oulasvirta. Social inference through technology. In *P. Markopoulos, B. De Ruyter, and W. Mackay (Eds.) Awareness Systems: Advances in Theory, Methodology and Design*, pages 125–147. Springer, London, 2009. doi: 10.1007/978-1-84882-477-5_5.

[132] K.-B. Park, S. H. Choi, J. Y. Lee, Y. Ghasemi, M. Mohammed, and H. Jeong. Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality. *IEEE Access*, 9:55448–55464, 2021. doi: 10.1109/ACCESS.2021.3071364.

[133] V. Pereira, T. Matos, R. Rodrigues, R. Nóbrega, and J. Jacob. Extended reality framework for remote collaborative interactions in virtual environments. In *2019 International Conference on Graphics and Interaction (ICGI)*, pages 17–24. IEEE, 2019. doi: 10.1109/ICGI47575.2019.8955025.

[134] C. Pidel and P. Ackermann. Collaboration in virtual and augmented reality: A systematic overview. In *L. T. De Paolis and P. Bourdot (Eds.) Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2020. Lecture Notes in Computer*

Science, vol. 12242, pages 141–156. Springer, Cham, 2020. doi: 10.1007/978-3-030-58465-8_10.

[135] J. Pirker. The potential of virtual reality for aerospace applications. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–8. IEEE, 2022. doi: 10.1109/AERO53065.2022.9843324.

[136] T. Piumsomboon, A. Day, B. Ens, Y. Lee, G. Lee, and M. Billinghurst. Exploring enhancements for remote mixed reality collaboration. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pages 1–5, Article 16, New York, NY, USA, 2017. ACM. doi: 10.1145/3132787.3139200.

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

[138] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, and M. Billinghurst. Mini-Me: An adaptive avatar for mixed reality remote collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, New York, NY, USA, 2018. ACM. doi: 10.1145/3173574.3173620.

[139] T. Piumsomboon, A. Dey, B. Ens, G. Lee, and M. Billinghurst. The effects of sharing awareness cues in collaborative mixed reality. *Frontiers in Robotics and AI*, 6:5, 2019. doi: 10.3389/frobt.2019.00005.

[140] T. Piumsomboon, G. A. Lee, A. Irlitti, B. Ens, B. H. Thomas, and M. Billinghurst. On the shoulder of the giant: A multi-scale mixed reality collaboration with 360 video sharing and tangible interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–17, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300458.

[141] S. Praharaj, M. Scheffel, H. Drachsler, and M. Specht. Literature review on co-located collaboration modeling using multimodal learning analytics—can we go the whole nine yards? *IEEE Transactions on Learning Technologies*, 14(3):367–385, 2021. doi: 10.1109/TLT.2021.3097766.

[142] Y. Y. Qian and R. J. Teather. 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, SUI '17, pages 91–98, New York, NY, USA, 2017. ACM. doi: 10.1145/3131277.3132182.

[143] J. Ren, Y. Weng, C. Zhou, C. Yu, and Y. Shi. Understanding window management interactions in AR headset + smartphone interface. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pages 1–8, New York, NY, USA, 2020. ACM. doi: 10.1145/3334480.3382812.

[144] T. Rhee, S. Thompson, D. Medeiros, R. dos Anjos, and A. Chalmers. Augmented virtual teleportation for high-fidelity telecollaboration. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1923–1933, 2020. doi: 10.1109/TVCG.2020.2973065.

[145] H. Ro, J.-H. Byun, Y. J. Park, N. K. Lee, and T.-D. Han. AR Pointer: Advanced ray-casting interface using laser pointer metaphor for object manipulation in 3D augmented reality environment. *Applied Sciences*, 9(15):3078, 2019. doi: 10.3390/app9153078.

[146] A. Rohacz, S. Weißenfels, and S. Strassburger. Concept for the comparison of intralogistics designs with real factory layout using augmented reality, slam and marker-based tracking. *Procedia CIRP*, 93:341–346, 2020. doi: 10.1016/j.procir.2020.03.039.

[147] M. Rudorfer, J. Guhl, P. Hoffmann, and J. Krüger. Holo Pick'n'Place. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1219–1222. IEEE, 2018. doi: 10.1109/ETFA.2018.8502527.

[148] A. H. Sadeghi, S. el Mathari, D. Abjigitova, A. P. M. Maat, Y. J. J. Taverne, A. J. C. Bogers, and E. A. Mahtab. Current and future applications of virtual, augmented, and mixed reality in cardiothoracic surgery. *The Annals of Thoracic Surgery*, 113 (2):681–691, 2022. doi: 10.1016/j.athoracsur.2020.11.030.

[149] A. Samini and K. L. Palmerius. A study on improving close and distant device movement pose manipulation for hand-held augmented reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 121–128, New York, NY, USA, 2016. ACM. doi: 10.1145/2993369.2993380.

[150] K. A. Satriadi, B. Ens, M. Cordeil, B. Jenny, T. Czauderna, and W. Willett. Augmented reality map navigation with freehand gestures. In *2019 IEEE Conference*

on Virtual Reality and 3D User Interfaces (VR), pages 593–603. IEEE, 2019. doi: 10.1109/VR.2019.8798340.

[151] A. Schäfer, G. Reis, and D. Stricker. A survey on synchronous augmented, virtual, and mixed reality remote collaboration systems. *ACM Comput. Surv.*, 55(6):1–27, Article 116, 2022. doi: 10.1145/3533376.

[152] B. Schneider, G. Sung, E. Chng, and S. Yang. How can high-frequency sensors capture collaboration? A review of the empirical links between multimodal metrics and collaborative constructs. *Sensors*, 21(24):8185, 2021. doi: 10.3390/s21248185.

[153] H. Schrom-Feiertag, G. Regal, J. Puthenkalam, and S. Suette. Immersive experience prototyping: Using mixed reality to integrate real devices in virtual simulated contexts to prototype experiences with mobile apps. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 75–81. IEEE, 2021. doi: 10.1109/ISMAR-Adjunct54149.2021.00025.

[154] V. Schwind, S. Mayer, A. Comeau-Vermeersch, R. Schweigert, and N. Henze. Up to the finger tip: The effect of avatars on mid-air pointing accuracy in virtual reality. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '18, pages 477–488, New York, NY, USA, 2018. ACM. doi: 10.1145/3242671.3242675.

[155] P. Schütt, M. Schwarz, and S. Behnke. Semantic interaction in augmented reality environments for microsoft hololens. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019. doi: 10.1109/ECMR.2019.8870937.

[156] H. Si-Mohammed, J. Petit, C. Jeunet, F. Argelaguet, F. Spindler, A. Évain, N. Roussel, G. Casiez, and A. Lecuyer. Towards BCI-based interfaces for augmented reality: Feasibility, design and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1608–1621, 2020. doi: 10.1109/TVCG.2018.2873737.

[157] S. Siltanen and H. Heinonen. Scalable and responsive information for industrial maintenance work: developing XR support on smart glasses for maintenance technicians. In *Proceedings of the 23rd International Conference on Academic Mindtrek*, AcademicMindtrek '20, pages 100–109, New York, NY, USA, 2020. ACM. doi: 10.1145/3377290.3377296.

[158] B. Spittle, M. Frutos-Pascual, C. Creed, and I. Williams. A review of interaction

techniques for immersive environments. *IEEE Transactions on Visualization and Computer Graphics*, 29(9):3900–3921, 2023. doi: 10.1109/TVCG.2022.3174805.

[159] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 265–272, USA, 1995. ACM Press/Addison-Wesley Publishing Co. doi: 10.1145/223904.223938.

[160] P. Stotko, S. Krumpen, M. B. Hullin, M. Weinmann, and R. Klein. SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 25 (5):2102–2112, 2019. doi: 10.1109/TVCG.2019.2899231.

[161] G. E. Su, M. S. Sunar, and A. W. Ismail. Device-based manipulation technique with separated control structures for 3D object translation and rotation in handheld mobile AR. *International Journal of Human-Computer Studies*, 141:102433, 2020. doi: 10.1016/j.ijhcs.2020.102433.

[162] H. B. Surale, A. Gupta, M. Hancock, and D. Vogel. TabletInVR: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–13, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300243.

[163] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM. doi: 10.1145/1476589.1476686.

[164] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt. Augmented reality and robotics: A survey and taxonomy for AR-enhanced human-robot interaction and robotic interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–33, Article 553, New York, NY, USA, 2022. ACM. doi: 10.1145/3491102.3517719.

[165] J. Sweller, P. Ayres, and S. Kalyuga. *Cognitive Load Theory*. Springer, 2011.

[166] M. Tanaya, K. Yang, T. Christensen, S. Li, M. O'Keefe, J. Fridley, and K. Sung. A framework for analyzing AR/VR collaborations: An initial result. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 111–116. IEEE, 2017. doi: 10.1109/CIVEMSA.2017.7995311.

[167] T. Teo, A. F. Hayati, G. A. Lee, M. Billinghurst, and M. Adcock. A technique for mixed reality remote collaboration using 360 panoramas in 3D reconstructed scenes. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, pages 1–11, Article 23, New York, NY, USA, 2019. ACM. doi: 10.1145/3359996.3364238.

[168] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock. Mixed reality remote collaboration combining 360 video and 3D reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–14, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300431.

[169] T. Teo, M. Norman, G. A. Lee, M. Billinghurst, and M. Adcock. Exploring interaction techniques for 360 panoramas inside a 3D reconstructed scene for mixed reality remote collaboration. *Journal on Multimodal User Interfaces*, 14:373–385, 2020. doi: 10.1007/s12193-020-00343-x.

[170] H. Tian, G. A. Lee, H. Bai, and M. Billinghurst. Using virtual replicas to improve mixed reality remote collaboration. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2785–2795, 2023. doi: 10.1109/TVCG.2023.3247113.

[171] G. Tsamis, G. Chantziaras, D. Giakoumis, I. Kostavelis, A. Kargakos, A. Tsakiris, and D. Tzovaras. Intuitive and safe interaction in multi-user human robot collaboration environments through augmented reality displays. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 520–526. IEEE, 2021. doi: 10.1109/RO-MAN50785.2021.9515474.

[172] C. Tudor. The impact of the COVID-19 pandemic on the global web and video conferencing SaaS market. *Electronics*, 11(16):2633, 2022. doi: 10.3390/electronics11162633.

[173] Turbosquid. URL `https://www.turbosquid.com`.

[174] A. E. Unlu and R. Xiao. PAIR: Phone as an augmented immersive reality controller. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21, pages 1–6, Article 27, New York, NY, USA, 2021. ACM. doi: 10.1145/3489849.3489878.

[175] R. Vanukuru and E. Yi-Luen Do. Exploring the use of mobile devices as a bridge for cross-reality collaboration. In *2023 IEEE International Symposium on Mixed*

and Augmented Reality Adjunct (ISMAR-Adjunct), pages 41–43. IEEE, 2023. doi: 10.1109/ISMAR-Adjunct60411.2023.00016.

[176] R. Vanukuru, S. C.-C. Weng, K. Ranjan, T. Hopkins, A. Banic, M. D. Gross, and E. Y.-L. Do. DualStream: Spatially sharing selves and surroundings using mobile devices and augmented reality. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 138–147. IEEE, 2023. doi: 10.1109/ISMAR59233.2023.00028.

[177] C. Vazquez, N. Tan, and S. Sadalgi. Home Studio: A mixed reality staging tool for interior design. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, pages 1–5, Article 431, New York, NY, USA, 2021. ACM. doi: 10.1145/3411763.3451711.

[178] VDI-5200-1. Factory planning - planning procedures, 2011.

[179] S. Voelker, S. Hueber, C. Holz, C. Remy, and N. Marquardt. GazeConduits: Calibration-free cross-device collaboration through gaze and touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–10, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376578.

[180] J. von Willich, M. Schmitz, F. Müller, D. Schmitt, and M. Mühlhäuser. Podoportation: Foot-based locomotion in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–14, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376626.

[181] O. Špakov, H. Istance, K.-J. Räihä, T. Viitanen, and H. Siirtola. Eye gaze and head gaze in collaborative games. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pages 1–9, Article 85, New York, NY, USA, 2019. ACM. doi: 10.1145/3317959.3321489.

[182] P. Wacker, O. Nowak, S. Voelker, and J. Borchers. ARPen: Mid-air object manipulation techniques for a bimanual AR system with pen & smartphone. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–12, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300849.

[183] P. Wang, S. Zhang, X. Bai, M. Billinghurst, W. He, S. Wang, X. Zhang, J. Du, and Y. Chen. Head pointer or eye gaze: Which helps more in MR remote collaboration? In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1219–1220. IEEE, 2019. doi: 10.1109/VR.2019.8798024.

[184] T. Wells and S. Houben. CollabAR - investigating the mediating role of mobile AR interfaces on co-located group collaboration. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376541.

[185] M. Whitlock, E. Harnner, J. R. Brubaker, S. Kane, and D. A. Szafir. Interacting with distant objects in augmented reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 41–48. IEEE, 2018. doi: 10.1109/VR.2018.8446381.

[186] J. Wieland, J. Zagermann, J. Müller, and H. Reiterer. Separation, composition, or hybrid? – comparing collaborative 3D object manipulation techniques for handheld augmented reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 403–412. IEEE, 2021. doi: 10.1109/ISMAR52148.2021.00057.

[187] H.-P. Wiendahl, J. Reichardt, and P. Nyhuis. *Handbuch Fabrikplanung - Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten*. Hanser, 2009.

[188] J. Wolfartsberger. Analyzing the potential of virtual reality for engineering design review. *Automation in Construction*, 104:27–37, 2019. doi: 10.1016/j.autcon.2019.03.018.

[189] H.-T. Wu, W.-D. Yu, R.-J. Gao, K.-C. Wang, and K.-C. Liu. Measuring the effectiveness of VR technique for safety training of hazardous construction site scenarios. In *2020 IEEE 2nd International Conference on Architecture, Construction, Environment and Hydraulics (ICACEH)*, pages 36–39. IEEE, 2020. doi: 10.1109/ICACEH51803.2020.9366218.

[190] W. Xu, H.-N. Liang, Y. Zhao, D. Yu, and D. Monteiro. DMove: Directional motion-based interaction for augmented reality head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–14, New York, NY, USA, 2019. ACM. doi: 10.1145/3290605.3300674.

[191] J. Yin, C. Fu, X. Zhang, and T. Liu. Precise target selection techniques in handheld augmented reality interfaces. *IEEE Access*, 7:17663–17674, 2019. doi: 10.1109/ACCESS.2019.2895219.

[192] D. Yu, W. Jiang, C. Wang, T. Dingler, E. Velloso, and J. Goncalves. ShadowDancXR: Body gesture digitization for low-cost extended reality (XR) headsets.

In *Companion Proceedings of the 2020 Conference on Interactive Surfaces and Spaces*, ISS Companion '20, pages 79–80, New York, NY, USA, 2020. ACM. doi: 10.1145/3380867.3426222.

[193] F. Zaman, C. Anslow, A. Chalmers, and T. Rhee. MRMAC: Mixed reality multi-user asymmetric collaboration. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 591–600. IEEE, 2023. doi: 10.1109/ISMAR59233.2023.00074.

[194] S. Zhai, P. Milgram, and W. Buxton. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 308–315, New York, NY, USA, 1996. ACM. doi: 10.1145/238386.238534.

[195] L. Zhao, Y. Liu, D. Ye, Z. Ma, and W. Song. Implementation and evaluation of touch-based interaction using electrovibration haptic feedback in virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 239–247. IEEE, 2020. doi: 10.1109/VR46266.2020.00043.

[196] F. Zhu and T. Grossman. BISHARE: Exploring bidirectional interactions between smartphones and head-mounted augmented reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–14, New York, NY, USA, 2020. ACM. doi: 10.1145/3313831.3376233.

[197] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. Augmented reality for construction site monitoring and documentation. *Proceedings of the IEEE*, 102(2):137–154, 2014. doi: 10.1109/JPROC.2013.2294314.

Academic CV

Vera Marie Memmesheimer

Education

10/2018 – 09/2020	Master of Science in Socioinformatics Department of Computer Science University of Kaiserslautern
10/2015 – 08/2018	Bachelor of Science in Socioinformatics Department of Computer Science University of Kaiserslautern

Employment

since 10/2020	Research Associate at Human Computer Interaction Lab Department of Computer Science RPTU University Kaiserslautern-Landau
10/2017 – 09/2020	(Under)graduate Assistant Department of Computer Science University of Kaiserslautern

Experiences Abroad

03/2023 – 04/2023	Research Stays
and 04/2022 – 06/2022	Department of Mechanical and Aerospace Engineering University of California Davis, USA
09/2018 – 01/2019	Erasmus Student Exchange Programme, Master Studies Universidad Politécnica de Madrid, Spain

Publications

- **V. M. Memmesheimer**, M. Klar, H. Subbaraj, B. Ravani, J. C. Aurich, and A. Ebert (2025): Applying Consistent Spatial Interaction Techniques to Factory Layout Planning. In J. Y. C. Chen and G. Fragomeni (Eds.) Virtual, Augmented and Mixed Reality. HCII 2025. Lecture Notes in Computer Science, vol. 15788, pp. 93-112. Springer, Cham. doi: 10.1007/978-3-031-93700-2_7.

- L. Zaina, J. C. Campos, D. Spano, K. Luyten, P. Palanque, G. van der Veer, A. Ebert, S. R. Humayoun, **V. Memmesheimer** (Eds.) (2025): Engineering Interactive Computer Systems. EICS 2024 International Workshops. Lecture Notes in Computer Science, vol. 15518. Springer, Cham. doi: 10.1007/978-3-031-91760-8.

- **V. M. Memmesheimer**, K. J. Klingshirn, C. Herold, B. Ravani, and A. Ebert (2024): Move'n'Hold Pro: Consistent Spatial Interaction Techniques for Object Manipulation with Handheld and Head-mounted Displays in Extended Reality. In Proceedings of the European Conference on Cognitive Ergonomics 2024 (ECCE '24). Article 10, pp. 1-8. ACM, New York, NY, USA. doi: 10.1145/3673805.3673814.

- **V. M. Memmesheimer**, S. M. Schwenkreis, and A. Ebert (2024): Towards Enhanced User Representations for Handheld Mixed Reality. In Mensch und Computer 2024 – Workshopband. Gesellschaft für Informatik e.V. doi: 10.18420/muc2024-mci-ws06-205.

- **V. M. Memmesheimer**, I. T. Chuang, B. Ravani, and A. Ebert (2024). Mixed Reality Handheld Displays for Robot Control: A Comparative Study. In I. L. Nunes (Eds.) Human Factors and Systems Interaction. AHFE (2024) International Conference. AHFE Open Access, vol. 154. AHFE International, USA. doi: 10.54941/ahfe1005380.

- **V. M. Memmesheimer**, J. Löber, and A. Ebert (2024): AWARE^{S}CUES: Awareness Cues Scaling with Group Size and Extended Reality Devices. In J. Y. C. Chen and G. Fragomeni (Eds.) Virtual, Augmented and Mixed Reality. HCII 2024. Lecture Notes in Computer Science, vol. 14706, pp. 44–59. Springer, Cham. doi: 10.1007/978-3-031-61041-7_4.

- G. C. Van Der Veer, S. R. Humayoun, **V. M. Memmesheimer**, and A. Ebert (2024): Experience 2.0 and Beyond - Engineering Cross Devices and Multiple Realities. In Companion Proceedings of the 16th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '24 Companion), pp. 108–110. ACM, New York, NY, USA. doi: 10.1145/3660515.3662838.

- **V. M. Memmesheimer**, K. J. Klingshirn, B. Ravani, and A. Ebert (2023): Move'n'Hold: Scalable Device-Based Interaction for Mixed Reality Handheld Displays. In Proceedings of the European Conference on Cognitive Ergonomics 2023 (ECCE '23). Article 13, pp. 1-8. ACM, New York, NY, USA. doi: 10.1145/3605655.3605656.

- **V. M. Memmesheimer** and A. Ebert (2023): A Human-Centered Framework for Scalable Extended Reality Spaces. In J. C. Aurich, C. Garth, and B. S. Linke (Eds.) Proceedings of the 3rd Conference on Physical Modeling for Virtual Manufacturing Systems and Processes (IRTG 2023), pp. 111–128. Springer, Cham. doi: 10.1007/978-3-031-35779-4_7.

- **V. M. Memmesheimer** and A. Ebert (2022): Scalable Extended Reality: A Future Research Agenda. Big Data and Cognitive Computing, 6(1):12. doi: 10.3390/bdcc6010012.

- **V. M. Memmesheimer** and A. Ebert (2022): Towards Advanced Evaluation of Collaborative XR Spaces. In C. Ardito et al. (Eds.) Sense, Feel, Design. INTERACT 2021. Lecture Notes in Computer Science, vol. 13198, pp. 443–452. Springer, Cham. doi: 10.1007/978-3-030-98388-8_40.

- A.-P. Lohfink, **V. M. Memmesheimer**, F. Gartzky, and C. Garth (2021): The Enhanced Security in Process System – Evaluating Knowledge Assistance. In IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX), pp. 1-7. IEEE. doi: 10.1109/TREX53765.2021.00006.