Using Augmented Reality in Everyday Life

Eugénie Brasier

université PARIS-SACLAY

Using Augmented Reality in Everyday Life

Usages de la Réalité Augmentée pour la Vie Quotidienne

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire interdisciplinaire des sciences du numérique (Université Paris-Saclay, CNRS)**, sous la direction de **Caroline APPERT**, Directrice de recherche

Thèse soutenue à Paris-Saclay, le 13 décembre 2021, par

Eugénie BRASIER

Composition du jury

Wendy MACKAY Directrice de recherche, Inria Paris-Saclay	Présidente
Laurent GRISONI Professeur, Université de Lille	Rapporteur & Examinateur
Laurence NIGAY Professeure, Université Grenoble Alpes (UGA)	Rapporteur & Examinatrice
Mathieu NANCEL Chargé de recherche, Inria Lille Nord - Europe	Examinateur
Caroline APPERT Directrice de recherche, LISN, Université Paris-Saclay, CNRS, Inria, Orsay, France	Directrice de thèse

THESE DE DOCTORAT
NNT : 2021UPASG110

ABSTRACT

Augmented Reality (AR) has a great potential of becoming a pervasive tool in users' everyday life. The technology is progressing rapidly and the research community is actively working on studying its uses, promising a future where users could take advantage of AR in their daily lives. Such a wide adoption of AR raises important questions regarding users' experience. This thesis identifies three specific problems regarding the usability of AR in everyday use. The first research question looks at low-level aspects of interaction with respect to users' motor performance and comfort when manipulating virtual content displayed in AR. We propose *ARPads* as a way to enable indirect input and thus reducing users' fatigue without losing speed and precision in comparison with direct manipulation of AR content. The second research question is about the integration of wearable AR within users' existing panorama of digital technologies. We focus on the combination of AR with handheld devices and propose a design space for enhancing phone-centric interactions with AR as well as quantitative results regarding the gained performance in the specific case of AR-enhanced widgets for phones. The last research question investigates high-level usability aspects when AR is always-on and thus can interfere with users' physical environment and social interactions. We introduce the concept of *AR de-augmentation*, giving users agency over the display of AR content to selectively remove specific AR content from their field of view.

FRENCH SUMMARY

Ce manuscrit présente les recherches que j'ai effectuées dans le cadre de ma thèse de doctorat autour des usages de la Réalité Augmentée (RA) dans la vie de tous les jours. J'y présente trois questions de recherche :

QR_1 Comment reconsidérer nos interactions avec la Réalité Augmentée portative pour que celles-ci soient appropriées pour des interactions prolongées ?

QR_2 Peut-on énoncer des recommandations sur l'augmentation des appareils mobiles avec la Réalité Augmentée portative ?

QR_3 Comment un utilisateur peut-il garder le contrôle sur le contenu qu'il perçoit en Réalité Augmentée ?

Le manuscrit est organisé en trois parties répondant respectivement à chacune de ces trois questions de recherche.

Pour répondre à notre première question (QR_1), nous présentons le projet ARPads. Un ARPad n'a pas de réalité physique, mais représente un plan d'interaction flottant dans l'air dans lequel les mouvements de l'utilisateur sont traduits par des mouvements de curseur dans une fenêtre de RA affichée dans l'environnement. Ce contrôle indirect permet à l'utilisateur d'adopter une posture plus confortable pour manipuler du contenu affiché en RA par opposition au contrôle direct qui force l'utilisateur à maintenir ses bras en l'air en direction du contenu de RA. Après avoir défini les critères de conceptions de ces ARPads en prenant en compte leur position et leur orientation relatives à l'utilisateur, nous avons implémenté et évalué certaines conceptions prometteuses. Nos résultats montrent que le contrôle indirect peut atteindre des performances similaires à celles du contrôle direct, tout en limitant la fatigue perçue par l'utilisateur et identifient les conceptions d'ARPads les plus efficaces.

Pour notre deuxième question (QR_2), nous nous concentrons sur la combinaison entre les appareils mobiles et la RA. Suite à un atelier de conception mené avec des utilisateurs d'appareils mobiles, nous avons défini un espace de conception d'applications centrées sur le smartphone pouvant bénéficier d'une association avec la RA. En particulier, nous avons

identifié deux dimensions principales : la fonction et la position du contenu holographique par rapport au contenu de premier intérêt affiché sur l'appareil mobile. Ce premier travail montre une grande richesse en termes de créativité pour proposer des améliorations possibles pour les appareils mobiles. Cependant, le manque de données quantifiables sur leur performance effective nous empêche encore de poser un avis objectif sur les réels avantages d'une telle association. Après avoir prototypé quelques exemples d'application pour démontrer leur faisabilité, nous nous sommes concentrés sur l'évaluation d'un cas spécifique : les composants d'interface (widgets) traditionnels des smartphones. En particulier, nous avons comparé différentes distributions de l'interface utilisateur entre la RA et le smartphone pour les widgets suivants : déclencheurs (boutons), ajustement paramétrique (slider) et sélection en 2 dimensions.

Enfin, pour notre troisième question (QR_3), nous nous projetons dans un futur dans lequel le champ de vision des utilisateurs contient fréquemment du contenu virtuel, et proposons une solution concrète pour que celui-ci n'interfère pas avec leur perception du monde réel. Pour cela, nous introduisons le concept de de-augmentation de la RA. Il s'agit de permettre aux utilisateurs de retirer volontairement certains éléments holographiques de leur champ de vision. Nous commençons par définir une taxonomie des opérations de dé-augmentation qui s'articule autour de trois dimensions : l'objet de la dé-augmentation, son déclencheur, et son rendu graphique. Nous illustrons ensuite ce concept au travers de trois scénarios démontrant l'utilité de telles opérations dans quelques situations particulières. Finalement, nous proposons une implémentation pratique de ce concept avec un prototype opérationnel, dont nous détaillons les différentes interactions qui permettent à l'utilisateur de définir une zone dé-augmentée. Ce dernier projet, plus théorique, ouvre des questionnements et des perspectives sur l'orientation des futurs systèmes de RA.

ACKNOWLEDGMENTS

Of course, my first acknowledgments go to my supervisor, Caroline Appert. Thank you Caroline for being an amazing supervisor, for your time, support, pedagogy, and most important, thank you for encouraging me to follow my own way at each step of this PhD.

Thank you to Laurence Nigay and Laurent Grisoni, which reviewed this thesis and provided me with great pieces of advice on how to make it better. They were also part of my defense's jury as well as Mathieu Nancel and Wendy Mackay, which presided over it. Thank you to the four of you for your attention, questions, and feedback.

Thank you to the other permanent members of the ILDA's team: Emmanuel Pietriga, Olivier Chapuis, and Anastasia Bezerianos. You were a listening ear when I needed so, you taught me that, even if I feel perfectionist when illustrating papers, worst people than me do exist on Earth. Most important, you have made my stay in this team a great experience.

I also want to thank all the old and current ILDA's members for having filled all the birthday cards, for having shot all the birthday videos, for having supported me in coding on the HoloLens, and, last but not least, thank you for having made this 3 years interesting, welcoming and (often) funny. In particular, thank you Olivier Gladin, Katia Evrat, Alexandra Merlin, Raphaël James, Emmanuel Courtoux, Mehdi Chakhchoukh, Vanessa Peña-Araya, Hugo Romat, Dylan Lebout.

Thank you to the interns I had the chance to work with (Mariam Diallo, Mamoun Elkhettar, Francesco Riccardo Di Gioia) who were all very motivated and reliable in their projects.

Thank you so much to Valentin Sauvage, who was my most important emotional support. He took part in several parts of this thesis, improving all projects I was invested in.

Thank you to my family (Christel, Joël, Justine, Clément, Margaux, and Hermes Brasier) that was another very significant support. I would also like to give a special thanks to my grandparents (Josette and Michel Deffrenne) who definitely helped me in popularizing my research topic.

Thank you to the FabLab Digiscop's staff that provided me with all the tools I needed to complete each of my projects.

Thank you to the head and administration staff of the doctoral school STIC for being very careful to the students' needs and for supporting me in my adventure in MT180.

CONTENTS

Part I

INTRODUCTION

When talking about Augmented Reality (AR), several images come to our minds. Movies [133, 80], Series [201], Anime [39], Books [194], Video Games [38] and even Art [1] have contributed to create a mental model of what AR is. Most people imagine users seeing holograms that they can interact with using mid-air gestures.

Besides this fictional vision, researchers have worked on providing a more formal definition of AR. In particular, Augmented Reality falls within the scope of Mixed Reality, which Milgram [128] has described as a continuum in 1994. However, this continuum, illustrated in Figure 0.1, does not specify clear boundaries of what AR encompasses, but broadly defines it as a technology that enhances the real environment with virtual information. When virtual elements are getting more important than real ones, Milgram no longer talks of AR but rather of Augmented Virtuality (AV). The use of a continuum suggests that there exist different levels of AR depending on the importance of the real environment in the system. Azuma [11] complements this definition by identifying three properties that characterize AR: (1) combining real and virtual, (2) being registered in three dimensions, and (3) being interactive in real-time.

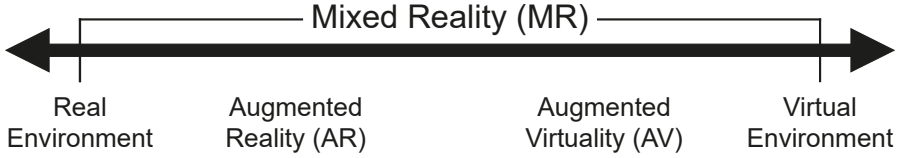


Fig. 0.1: Virtuality Continuum of Milgram [128]

AR is no longer part of fiction only. There is a range of technologies that can actually run AR systems. Among these, we can identify two broad categories depending on the technology's form factor: (1) *Stationary* setups and (2) *Mobile* setups. *Stationary* setups are integrated into users' environment and make it possible to augment a specific area. For example, Figure 0.2-a) illustrates *HoloDesk* [75], a system that projects a virtual layer onto a fixed transparent screen in front of the user. With such a system, users can interact with virtual elements by placing their hands under this transparent screen. *Mobile* setups are usually more lightweight than stationary setups. They are *portable*, making them better suited to diverse tasks and contexts. *Mobile* setups can be (1) *Handheld* or (2) *Wearable*. By definition, *Handheld* devices (*e.g.* tablet, smartphone) must be held. This not only occupies users' hands but also forces users to maintain a potentially uncomfortable posture. This makes these devices unsuitable for long-lasting tasks. For example, Figure 0.2-b) illustrates how the user should hold their smartphone to scan the real environment when using the application *Pokémon Go* [145]. On the opposite, *Wearable* devices, also called *Head-Mounted Displays* (HMD), can be permanently worn by users (*e.g.* headsets, glasses). For example, Microsoft has released multiple scenarios of AR use cases with their Hololens. Figure 0.2-c) illustrates such a scenario for a work environment where the user is manipulating a virtual object with two hands as they would do with a physical object.

a) b) c)

Fig. 0.2: The three forms factor of AR devices: a) *Stationary* AR setup (*e.g.* HoloDesk [75]), b) *Mobile* AR setup (*e.g.* Pokemon Go [145] using a smartphone), c) *Wearable* AR setup (*e.g.* Commercial for Microsoft HoloLens [127]).

Besides their form factor, *Wearable* AR devices can also be separated into two other categories, depending on their rendering method. On the one hand, Augmented Reality can be obtained using *Video See-Through* (VST) devices, which video record the real environment and modify the video stream to include virtual objects and thus render an augmented view to the user (*e.g.* Varjo - XR - 3 [203], smartphones). However, such devices raise an important issue: if the VST device stops working, the user gets completely blind. The second way of obtaining AR relies on *Optical See-Through* (OST) devices, which superimpose a virtual layer on the environment by projecting it on a transparent or semi-transparent screen (*e.g.* Google Glasses [58], Microsoft HoloLens [127]). Even if the performances of VST devices are higher, we strongly believe that OST devices have a great potential to get widely adopted in users' everyday life. Indeed, users are not completely cut off from their environment as these devices do not "replace" reality, but rather supplement it.

Looking at their history, they have evolved a lot and have recently started to get close to a technology that could actually get adopted at large. If OST devices still benefit from the novelty effect of new technologies, researchers have been actually studying them for a while. The first prototype of an AR *Optical See-Through* device has been presented in 1968 by Sutherland [197] (*cf.* Figure 0.3 - a)). Even if the system was crude, it already integrated specific sensors for tracking the user's head position and rotation, which were used to change the perspective of a virtual element displayed through cathode tubes and overlaid with reality thanks to a mirror-based system. Since 1968, there have been major improvements regarding AR devices' performances, not only regarding their tracking and rendering capabilities but also regarding their form factor. Indeed, while Sutherland's system could only be used within a specific room, Feiner *et al.* introduced a portable system called the *Touring Machine* [50] (*cf.* Figure 0.3 - b)) in 1997. It consisted of an OST tied to a backpack containing a full computer, providing users with virtual in-context annotations on real buildings when walking in their university campus. Electronic miniaturization then made it possible to reduce HMD's volume and weight, while significantly increasing their performances. Not only are they lighter

and more aesthetic, but they are faster and have a higher display resolution. These improvements have led to the Microsoft HoloLens 2 [127] (*cf.* Figure 0.3 - c)) in 2017. Current technologies such as the Microsoft HoloLens now allow users to interact with various modalities (*e.g.* gaze, touch, voice), while remaining relatively lightweight (566g). Their price also suggests that AR HMDs are getting more and more affordable [125]. However, this evolution has not yet driven to a final product for daily use. They are still quite uncomfortable and conspicuous. But, a future direction for AR devices includes metasurfaces such as contact lenses [150, 153, 136] (*cf.* Figure 0.3 - d)), promising a possible future where AR could be worn on an everyday basis. Yet, these are still in the early stages of development.

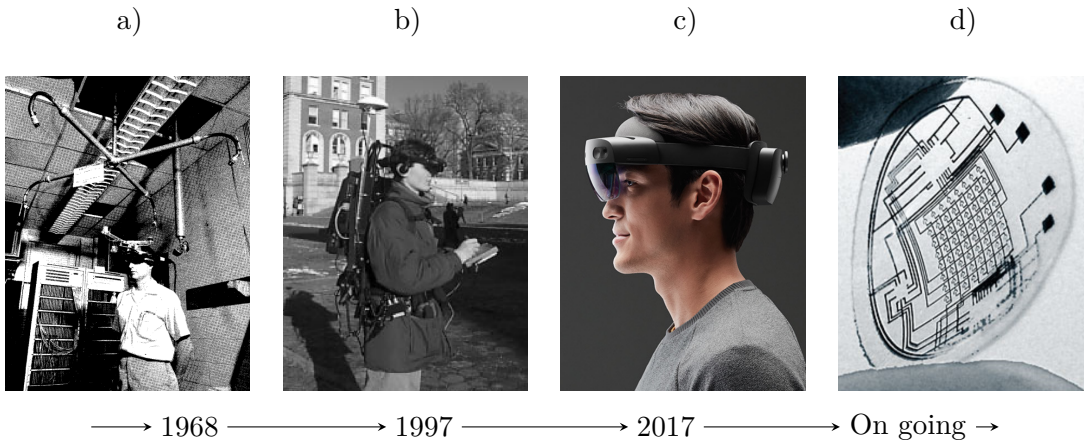


Fig. 0.3: Evolution of Augmented Reality devices. a) 1968 - Head-Mounted Display of Sutherland [197]; b) 1997 - the Touring Machine of Feiner *et al.* [50]; c) 2017 - Microsoft HoloLens [127]; d) On going - AR contact lenses [150]

Besides technological advances, some researchers also argue that Augmented Reality has the potential of becoming a pervasive tool in users' everyday life [125]. Indeed, it has already proved useful for specific use cases in both academia and industry. In the field of education, AR has the potential of providing a 3D immersive environment that facilitates learning with a higher recalling, understanding, and engagement [15, 169]. For example, it can be helpful for users' spatial orientation and their understanding of 3D relief [27]. The higher level of engagement can also profit culture at large by revitalizing historical elements [152] or enhancing art [1]. In Industry, AR can be used to support workers in assembly tasks [55].

Wearable AR can also assist people with some visual impairments. For example, the ChromaGlasses [104] proposes to correct images by exaggerating some colors in the AR virtual overlay to lower the mental demand of color blinds and increase their level of confidence while describing colors. Other types of overlays can also be considered for mitigating other visual disabilities. For example, an AR overlay can provide a magnifying

view of the scene, increase the visibility of obstacles with higher contrast, provide audio feedback of what is situated within the scene, *etc.* [131]. AR could also be used as a tool for therapy as it can bring new fully virtual elements in users' field of view (FoV), exposing users to *e.g.* a phobia, while feeling safer than in a real situation [87]. AR can also train people for specific situations in emergency management and clinical environment [178] as such situations could not be reproduced in real life. But AR is not limited to these specific uses and more casual use cases for daily life have been described in the literature too. AR can *e.g.* enhance gaming by providing real-life accessories and metaphors for interaction [157], making users more engaged. Some HCI researchers are also envisioning the use of AR in a large range of domestic contexts. For example, AR can assist in caring for a baby, cooking, alerting of dangerous elements in a room to prevent injuries, or enhancing home devices [97].

In summary, Augmented Reality has shown its usefulness in many fields, and promises a large range of yet non-explored other use cases for helping users with their daily lives. However, echoing Koelle [98], I argue that there is still a range of questions regarding the usability of AR to address before actually considering it as a viable technology for everyday use. I detail below the three research questions that my thesis addresses.

The first question is about the means of interacting with wearable AR. In most existing systems, users have to involve their bodies when interacting with virtual object as they would do when interacting with physical objects in a real life context (Figure 0.2-c) illustrates such a case). But AR is very different from reality as interactable elements are fully virtual and thus have no tangibility. The early stage of maturity of AR has encouraged the design of short-term interactions, mostly based on real life metaphors where users *touch* virtual objects. However, some studies suggest that large movements in the air are tiring and not suitable for continuous use [12, 76]. Furthermore, interacting with large gestures can be socially awkward, especially with personal devices, as external people cannot see the virtual content displayed in the personal view. Therefore, before AR gets widely adopted, there is a need for reconsidering interaction with wearable AR. My first research question is:

How to reconsider interaction with wearable AR to better support long-lasting tasks?

The second question revolves around the place of wearable AR within our already large set of digital devices. Like smartwatches have not replaced smartphones, wearable AR devices will likely co-exist with other digital devices. As Augmented Reality typically turns the air into an interactive space, it offers promising prospects for enhancing other devices that are constrained by their limited interactive space. Smartphones are especially good candidates for this as they have a very limited size while offering good computing performance, good input precision, and high display resolution. Normand *et al.* [148] gave a first overview of the possible smartphone-based applications that could benefit

from being combined with an AR headset. Yet, there are no clear guidelines for designers and developers about successful ways of combining a personal handheld device with a wearable AR device. My second research question is:

Can we provide guidelines on how a wearable AR device can enhance a handheld device?

Augmented Reality definitely has the potential of changing users' routines by blending the boundaries between real and virtual content [11]. Despite all its advantages, this technology is very intrusive as it almost directly modifies users' perception of their environment. Virtual and real environments are superimposed, which can create confusion for users due to visual interferences between the two environments. The last question regards how users could be able to differentiate real and virtual elements, be it for their safety or for ethical considerations. Before integrating AR in a daily context, there is a need to provide users with "safeguards", allowing them to take advantage of AR without being misled or endangered by it. My third research question is:

How can users be in control of what Augmented Reality tells them?

While these are broad and complex questions, this manuscript reports on projects I conducted during my PhD to provide preliminary answers. First, we investigate the concept of mid-air pads (*ARPads*) as an alternative to gaze or direct hand input to control a cursor in windows anchored in the environment. An *ARPad* is a virtual plane within which users can move their hands to control a cursor displayed in the headset screen. Contrary to direct input, an *ARPad* is not spatially aligned with the headset screen, meaning that users can perform mid-air movements in a more relaxed, comfortable posture. We investigate a design space for *ARPads*, which takes into account the position of the pad relative to the user's body, and the orientation of the pad relative to that of the headset screen. Our study suggests that 1) indirect input can achieve the same performance as direct input while causing less fatigue than hand raycast, 2) an *ARPad* should be attached to the wrist or waist rather than to the thigh, and 3) the *ARPad* and the screen should have the same orientation.

Then, I present an investigation of the combination of an AR headset with a handheld device. Following a user-centered approach, we started with a workshop that we organized with smartphone users to brainstorm about the possible enhancements that AR could bring to mobile devices. Our observations drove the definition of a design space of smartphone+AR applications based on two dimensions: the spatial location of the augmented content around the smartphone, and the function of that augmented content with respect to the primary content displayed on the phone. We then prototyped some applications to populate interesting cells of this design space with examples. Finally, in order to provide enhancements that are readily applicable to existing smartphone applications, we focused on existing smartphone widgets and designed several ways of

enhancing them with AR. We propose several solutions for manipulating discrete and continuous widgets (1D and 2D) with a varying amount of interaction offloaded in the air. We compare these solutions against their state-of-the-art touch-only counterparts with a series of exploratory studies in which participants had to perform three tasks: command trigger, parameter value adjustment, and precise 2D selection. We then derive guidelines from our empirical observations.

Finally, I introduce the concept of *De-augmenting* AR, which allows users to define arbitrary areas that are freed from augmented content. In the *De-augmenting* AR project, we rationalize the need to give users agency over the visibility of this virtual content. De-augmentations let users selectively remove AR elements from their field of vision, only targeting those that actually interfere with their perception of the physical world. We build a taxonomy that captures the different facets of de-augmentation, discuss a variety of scenarios that illustrate its usefulness, demonstrate its practical feasibility, and discuss multiple questions that it raises about the design of future AR systems.

As described above, this thesis offers interesting response elements over the proper way to integrate AR in users' everyday life. However many questions remains. Inspired by our work and the related litterature, I discuss some of these in a last chapter, putting forward directions for future research on this topic.

Part II

PROJECTS

1. MID-AIR INDIRECT INPUT FOR AUGMENTED REALITY

This chapter is based on the paper "ARPads: Mid-air Indirect Input for Augmented Reality" [18] that has been presented in ISMAR '20. The co-authors of this paper are Olivier Chapuis, Nicolas Ferey, Jeanne Vezien, and Caroline Appert.

Augmented Reality (AR) headsets make it possible for users to get private, in-situ user interfaces. They can offer personalized exhibits in museums [152], deliver specific instructions in an educational context [57], or display data visualizations related to objects in the environment [184]. Interacting with virtual objects displayed in the headset is typically achieved through gaze and hand gestures performed in front of the headset. For example, the Microsoft Hololens requires users to move their head to adjust the cursor position over the object of interest, and then raise their arm to perform an air-tap hand gesture in the tracking area located ahead of the headset.

Both gaze input and mid-air gestures have drawbacks. Mid-air gestures can quickly get tiring when the arms are in an upward position [12, 76], while gaze-based input suffers from precision issues [208] and is prone to accidental selections [83, 84]. In the case of AR headsets that estimate gaze based on head position and orientation, the situation is even worse as the object to select needs to be put in the viewport's center, making peripheral (but potentially interesting) objects leave the user's field of view. Furthermore, large gestures performed in mid-air can be socially awkward [2]. So can unnatural head movements. This might result in a mismatch between displayed content that is intended for private consumption and exaggerated movements to interact with this content.

We investigate the concept of *ARPads* as a means to replace or complement the default input channels of AR headsets. An *ARPads* can be seen as a virtual plane in which users perform small scale movements to control the cursor displayed in the headset's screen, like a trackpad does with a cursor displayed on a laptop's screen.[1] As already demonstrated in the context of remote control of distant vertical screens [113], this indirect way of controlling a cursor enables users to adopt more relaxed postures and perform gestures that are subtle, and thus more acceptable in public [47].

[1] The parallel is not to be considered literally as a trackpad enables relative cursor control (with clutching actions) while our implementation of an *ARPads* enables absolute control (as a graphics tablet).

We investigate a *design space* for such indirect input in the context of AR headsets. We consider two dimensions: where the gestures are performed relative to the user's body (wrist, waist or thigh), and what the main orientation of those gestures is (horizontal, vertical). We implement specific designs in this space in an AR headset that has a limited-size viewport (*i.e.*, a Microsoft Hololens 1). We conduct a study to evaluate these designs as well as two direct input baselines (gaze and hand raycast) on both discrete and continuous input tasks. Our study results indicate that: 1) indirect input can achieve the same performance as direct input while causing less fatigue than hand raycast; 2) an *ARPad* is more efficient when positioned close to users' wrist or waist rather than to their thigh; and 3) users perform better with an *ARPad* when its orientation matches that of the headset screen.

Our contributions are:

1. a design space mid-air indirect input techniques for AR;
2. a study of these input techniques compared to their state-of-the-art counterparts.

1.1 Related Work

Indirect input is a fundamental concept in HCI that received a lot of attention in the literature about pointing. We only discuss here on works that are closely related to our contribution, focusing on interaction with head-mounted displays and mid-air gestures for interacting with portable devices.

1.1.1 Interacting with Head-mounted Displays

Usability is a key aspect to consider if we want to facilitate the adoption of AR glasses [98]. Many projects in the literature aim at improving interaction with such devices. As mentioned above, interaction with AR headsets usually relies on gaze input. While fast, gaze input suffers from precision issues [208]. Gaze input thus benefits from the addition of other modalities [101] such as a mouse in a desktop setting [217] or a handheld touch device in a mobile setting [193]. Gaze input also suffers from the Midas Touch problem [83, 84], which can cause accidental activation commands in the absence of a proper selection mechanism. Selection with gaze is usually performed using a dwell action. It can also rely on an additional device [83, 208]. When targets are moving, selection can also be achieved with eye pursuit [95, 163].

Mid-air gestures have also received a lot of attention as a means to interact with head-mounted displays. They remove the need for additional devices and allow users to perform *direct manipulation* of objects that are within arm's reach [132]. Mid-air gestures can be complemented with another modality such as speech [162] or gaze [156]. Commercial solutions like the Microsoft Hololens rely on gaze for pointing and air tap gestures for selection. In this specific context of AR headsets, optical sensors for tracking hand gestures are usually mounted on the headset, looking in the direction of users' sight (*e.g.*, [36, 68]). This means that users have to perform gestures at a relatively high height, which quickly causes discomfort and fatigue [76, 158].

When users do not need to keep their hands free, an external device can be used as a controller for Head-mounted Displays. This external device can be a smartwatch [159], a smartphone [23], a tablet [196], or a custom one [177]. The tangibility of an external device usually improves comfort and precision. For instance, a smartphone can act as a trackpad for pointing content displayed on a distant screen [186] or in AR [35]. A spatially tracked mobile device can control 6 degrees of freedom (DOF) for manipulating spatial content [130, 135]. Further adding multitouch input, users can even perform advanced manipulations in VR [196]. Conversely, while portable devices can address weaknesses of HMDs in terms of input, HMDs can also address weaknesses of portable devices in terms of output [218, 62]. For instance, an AR display can enlarge a smartphone's display to accommodate a large information space [148] or to distribute widgets across displays [60].

For tasks performed in the office, users can lay a tablet on their desk and use its soft keyboard for text entry in VR [64]. They can also use a mouse [57] or a physical keyboard [64]. In a mobile context, a smartwatch can be used to interact with remote content: for raycast pointing [159], for indirect pointing [213, 218] or for text input [3]. All these projects advocate using an external input device to interact with HMDs. We argue that bare-hand input can complement such techniques or replace them when an external device is not available. Bare-hand input also has the advantage of making HMDs stand-alone devices, and of leaving users' hands free for other actions.

A hands-free alternative consists in using the user's body as an input device with large-amplitude gestures performed on the whole body [54] or with finger gestures performed on specific parts of the body (*e.g.*, on the forearm [71], on the hand palm [140], on the abdomen [205], and even on the face [107, 181, 214]). Wearable devices also support portability. Dobbelstein *et al.* have proposed wearables for interacting with headset screens with, *e.g.*, a touch-enabled belt [40] and pocket [41]. Finally, foot taps have also been investigated recently as a means to perform menu selection with head-mounted displays [141]. However, these latter solutions require custom sensors, have limited input capabilities and might raise questions regarding social acceptance.

1.1.2 Overcoming Fatigue of Mid-air Gestures

In light of the above, mid-air gestures remain a very good candidate for interacting with headset screens. However, tracking such gestures in a portable setting is not trivial. Some research prototypes embed cameras in clothes, shoes or jewels. For example, the Gesture Pendant [192] is a wearable device used to control home environments with mid-air gestures. SixthSense [134] extends the Gesture Pendant by combining a projector and a camera in a pendant. It can project digital content in the physical environment in front of the user, and capture gestures that are performed at chest height. ShoeSense [13] augments a shoe with a Leap Motion to track large-amplitude arm and hand gestures to control wearable applications. Such wearable tracking equipment has the advantage of allowing users to perform mid-air gestures with arms in a position lower than when optical sensors are mounted on the headset itself.

OwnerShift [52] is a recent technique for Virtual Reality (VR) headsets that enables users to progressively relax their posture during long-lasting interactions. Two optical sensors are mounted on the headset itself, one oriented along users' sight, the other oriented downward. Users see a virtual representation of their hands, which gets progressively shifted up so as to encourage users to shift their real hands down, smoothly transitioning from overhead to waist. OwnerShift builds upon the set of techniques that play with the rubber hand illusion [17]. For example, the Go-go Interaction Technique virtually extends users' arms to reach distant objects without walking [166]. The converse approach also works: instead of virtually extending users' body, virtual objects can be brought closer in motor space while preserving their positions in visual space [137]. However, AR gives less freedom than VR as it makes it challenging to modify the representation of users' body.

Gunslinger [113] addresses the fatigue problem of mid-air gestures in a more radical manner. Relying on what has been later called *at-your-side gestures* [185] and earlier used for zooming actions in [143], Gunslinger allows users to remote control a large display using small-scale movements that are performed in a relaxed posture. One Leap Motion controller is attached to each thigh, tracking both hands when arms are along the body. Users point with their dominant hand following a trackpad metaphor, and activate commands with specific postures of their non-dominant hand. Gunslinger's support for pointing is very similar to a horizontal *ARPad* attached to users' thigh in our design space, with one notable difference: Gunslinger works in relative mode (with clutching actions) while *ARPads* work in absolute mode.

1.1.3 Extending Input Space in the Air

Extending the interaction space in the air is a strategy that has been implemented for devices other than head-mounted ones. It was initially investigated for small devices because of the limited size of their interaction surface, which causes occlusion and precision issues. For example, the Gesture Watch [96] allows users to perform mid-air gestures above the watch. Gestures are performed with the dominant hand over the watch worn on the non-dominant arm (the watch is equipped with an array of proximity sensors). The Abracadabra wristwatch [70] extends a watch's sensing area above the surface in the same way, but differs in the sensing technology: the finger from the dominant hand wears a magnet, making it trackable by the watch's magnetometers. Here again, those techniques share some similarities with *ARPads* when they are located on users' wrist with a horizontal orientation.

Building upon the concept of *continuous interaction space*, that considers a touch surface and the space around it as a continuum [119], Chen *et al.* equip a smartphone with a depth camera in order to extend the vocabulary of possible commands with air+touch gestures [34]. The combination of touch and mid-air input has also proven efficient for performing large amplitude slide gestures on a smartphone [8] or on a smartwatch [69].

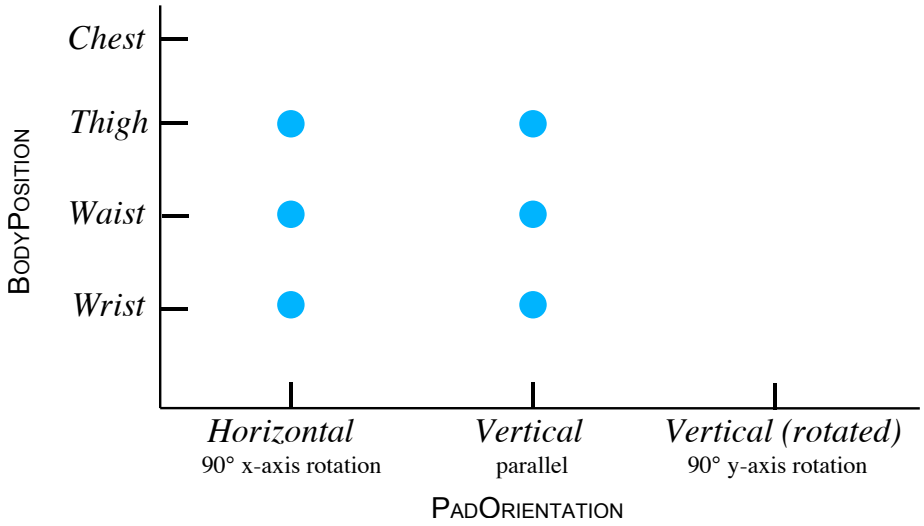


Fig. 1.1: Our design space is structured along two dimensions: where the *ARPad* is located relative to the user's body (BODYPOSITION), and how the *ARPad* is oriented (PADORIENTATION). Blue points indicate the six specific configurations that we study.

1.2 ARPad Design Space

Mid-air indirect input, as a concept, is not novel. But all possible designs have not been systematically investigated. This section describes our *design space* for *ARPads*. This space encompasses techniques from the literature, as well as novel ones that build upon prior work on mid-air gestures for portable devices.

An *ARPad* is an imaginary pad on which users can move their hands to control the position of a cursor displayed in the headset screen. Like imaginary interfaces [67], users operate *ARPads* with their bare hands and without any visual support. However, *ARPads* differ from the concept envisioned by Gustafson *et al.* where imaginary *interfaces* are operated with direct hand gestures. An *ARPad* is rather an imaginary *device*, which is used for indirect control of a distant screen. The number of possibilities for implementing such a mid-air pad is large. Our design space, illustrated in Figure 1.1, focuses on two dimensions that relate to the positioning of *ARPads*. PADORIENTATION describes the *ARPad*'s position relative to the AR screen as it is key to the definition of indirection. BODYPOSITION describes the *ARPad*'s position relative to the user's body as ergonomic aspects strongly depend on it.

The first dimension, BODYPOSITION, is directly motivated by previous work on mid-air gestures for mobile contexts (discussed in the previous section). Tracking mid-air gestures while leaving users free to move in their environment requires attaching optical sensors to them. Depending on which body part the sensors are attached to, users will be able

to perform gestures at different locations relative to their body. In our design space, dimension BODYPOSITION represents the body part close to which users perform mid-air gestures. Here, we specifically consider three body parts: *Thigh*, *Waist* and *Wrist* (Figure 1.2). *Thigh* is an obvious candidate as it has already been used for cursor control in the Gunslinger project [113]. Although not focusing on indirect pointing, mid-air gestures relative to users' *Wrist* have also been considered for improving interaction with smartwatches (*e.g.*, [70, 96]). In projects where mid-air gestures were tracked by a pendant [192, 134], users were performing gestures in front of their *Chest*. However, considering empirical results about the fatigue of mid-air gestures when arms are held high in the air [76], *Waist* is a more promising option to study. When gestures are tracked at waist level, users can keep their arms low and should experience less fatigue.

The second dimension, PADORIENTATION, describes the orientation of the 2D plane within which the pad stands. We consider *Vertical* pads, which correspond to cases where the *ARPad*'s orientation matches that of the headset screen when users look straight ahead. We also consider *Horizontal* pads, which correspond to cases where the *ARPad* is rotated 90° around the x-axis of the headset screen. This configuration is particularly interesting, as we can expect it to be familiar to users who have experience with a physical trackpad on a laptop computer. While we limit our first investigation to these two specific values, other values might be worth investigating. For example, for an *ARPad* located close to users' *Thigh*, we could also consider the case where the *ARPad* is rotated 90° around the y-axis of the headset screen in order to allow for movements parallel to the sagittal plane that might be comfortable. We could even go beyond 2D planes and consider curved surfaces that might better fit the trajectory of limbs around joints.

Figure 1.3-left shows where the six *ARPad* designs that we investigate stand in our design space. These designs result from crossing values {*Thigh*, *Waist*, *Wrist*} for BODYPOSITION with values {*Horizontal*, *Vertical*} for PADORIENTATION.

1.3 ARPad Prototypes

We implemented the six *ARPads* identified above using a Microsoft HoloLens and a Leap Motion controller. We relied on Unity (2017.2.4.f1) and the MixedRealityToolkit (2017.4.3.0) to implement our own AR cursor. We reused the `IPointingSource` interface from MRTK, and registered the cursor in the `FocusManager` instance as the main one. The pointer position gets updated each frame depending on what has been sensed with the Leap Motion (without being affected by head motion). Object picking in the AR scene is achieved through raycasting from this pointer.

We rely on a Leap Motion controller for tracking users' hands. Depending on the pad's orientation (BODYPOSITION), we attach the Leap Motion to users with pieces of velcro tape, as illustrated in Figure 1.2. As opposed to Gunslinger [113], which implements

Fig. 1.2: Waist, *Wrist* and *Thigh* body positions for the Leap Motion

relative control with clutching postures, *ARPads* work in absolute mode. We made this choice in order to focus on motor and cognitive aspects involved in indirect control for AR, eliminating any potential issue related to a recognition engine for hand postures. Dimensions of an *ARPad* are 10.3 × 18.3 cm. The LeapMotion provides a 3D coordinate system whose origin is the device itself. We transform coordinates from this cuboid to the *ARPad*'s plane by ignoring the axis that is orthogonal to that plane. This means that an *ARPad* is a cuboid inside which all movements are projected on a plane rather than an actual plane on which users would have to perform unnatural co-planar movements. Informed by pilot studies, we implemented a control display ratio of 1cm:1.38°, which is uniform on both the x- and y-axes (Figure 1.4) . This means that a 1cm hand movement will make the cursor move 1.38° in visual angle.

We use an external computer for communication between the Leap Motion and the HoloLens. As the Leap Motion cannot be directly plugged to the HoloLens, it sends data to the computer via USB, which in turn sends them to the HoloLens via sockets.

1.4 Experiment

Our study aims at assessing 1) how indirect input performs in comparison with direct input, and 2) what *ARPad* designs perform best.

1.4.1 Direct Input Baselines

HeadGaze – An AR headset cursor is usually controlled through gaze, allowing users to directly look at the object they want to select. In current headsets such as the Microsoft HoloLens, gaze orientation is actually estimated based on head orientation. Our first baseline (*HeadGaze*) corresponds to this default cursor control technique.

HeadHand – In our second baseline, users directly control the cursor with their hand. The cursor position results from a raycast along the vector headset-hand. As opposed to indirect techniques, *HeadHand* implements a 1:1 control display ratio to make the cursor position match that of the hand. This corresponds to the *Image-Plane* technique in [86].

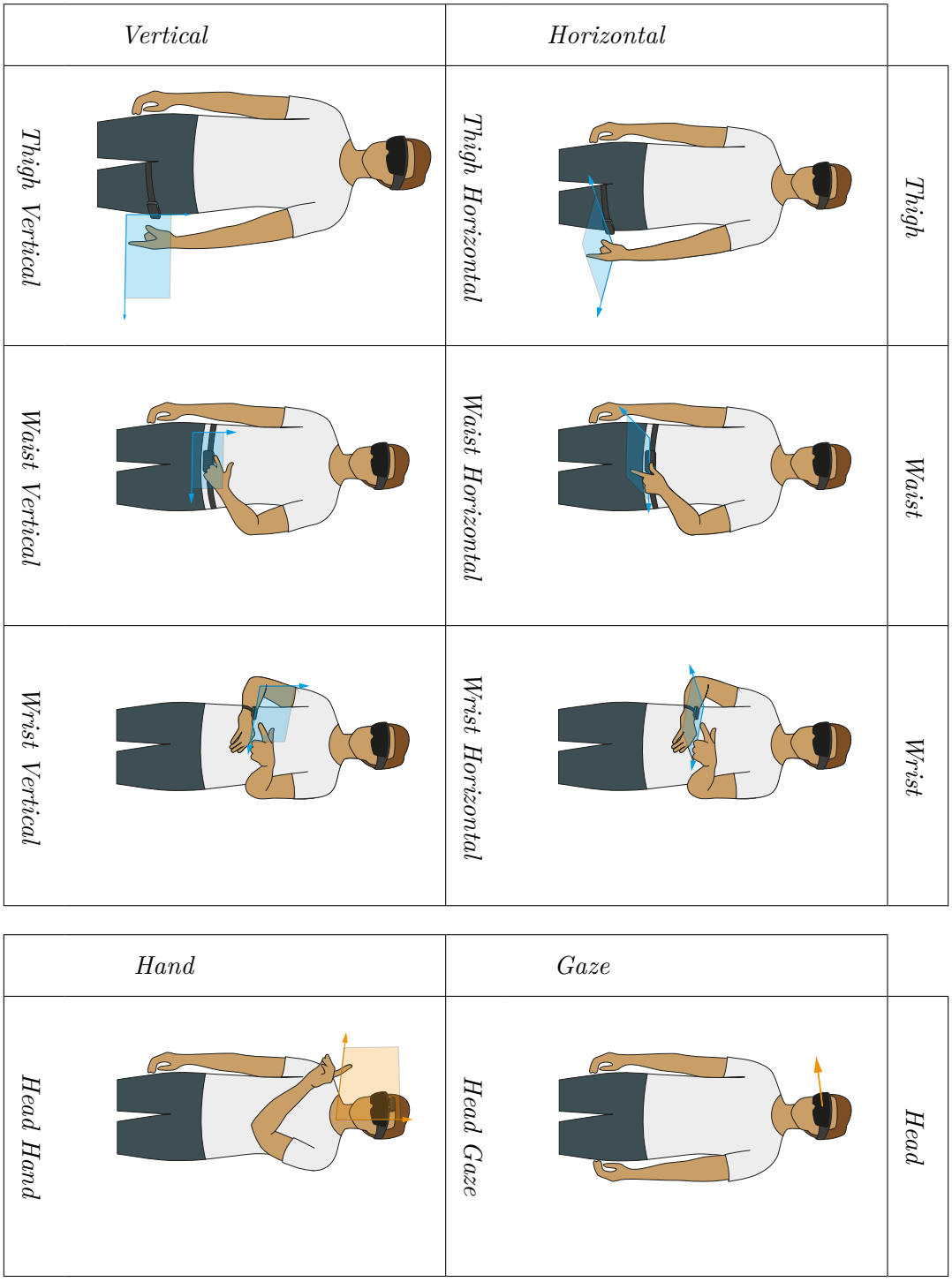


Fig. 1.3: (Left) Indirect input conditions: the six *ARPad* specific configurations from our design space $\{Thigh, Waist, Wrist\} \times \{Horizontal, Vertical\}$. (Right) Direct input conditions: *Gaze* and *Hand* raycast.

.

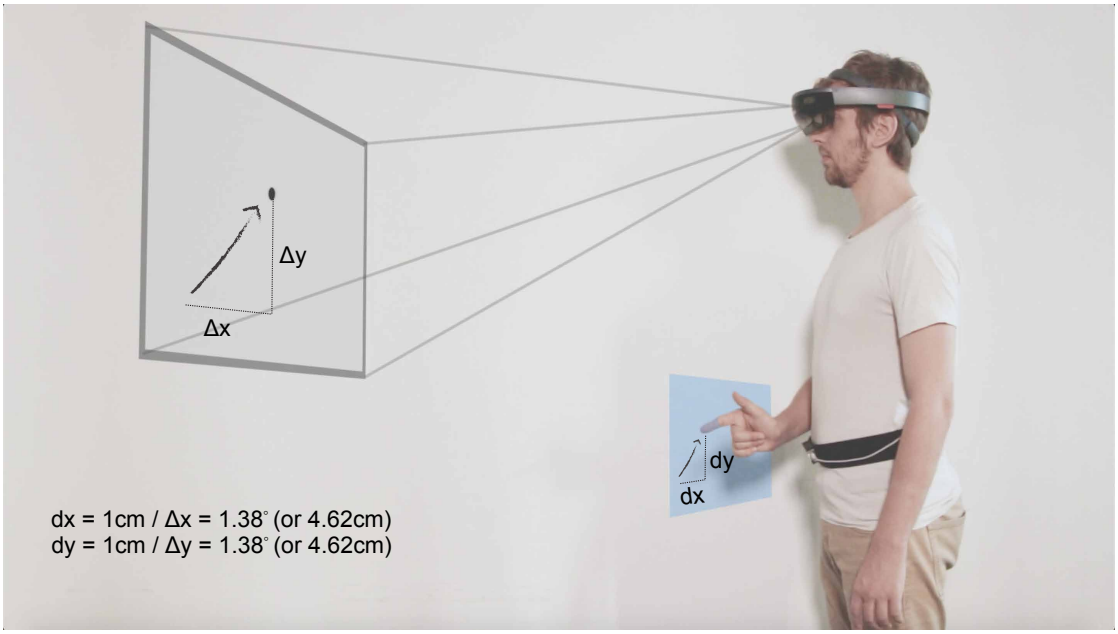


Fig. 1.4: Control display ratio for controlling a cursor displayed in the headset screen through movements on the *ARPad*.

1.4.2 Hypotheses

Our experiment compares the eight techniques illustrated in Figure 1.3. Our hypotheses regarding their comparative performance were as follows:

H_1 *Users experience more fatigue with HeadHand than with ARPads.* This first hypothesis is the actual motivation for using indirect input as an alternative to hand direct input. Freehand interaction when holding arms up in the air are tiring [76]. *ARPads* let users keep their arms in positions that are more comfortable than *HeadHand* does.

H_2 *Users perform better with direct techniques than with ARPads.* Visual and motor spaces are not spatially aligned with *ARPads*. Such an indirection can degrade user performance, as is the case for a typing task on a software keyboard in Virtual Reality [64]. We expect that this indirection will also be cognitively demanding when pointing with an *ARPad*, and that it may affect user performance.

H_3 *Vertical ARPads perform better than horizontal ones.* The level of indirection between the motor space and the visual space is even higher when both spaces differ in their orientation. This might degrade user performance even more.

H_4 *An ARPad at Wrist position performs better than at Waist or Thigh positions.* When the *ARPad* is located on the *Wrist*, users can move their non-dominant hand to adjust the plane's location and orientation. This enables bimanual actions where the non-dominant hand sets the context for precise actions performed with the dominant one. Such an asymmetric division of labor might prove efficient [66].

1.4.3 Participants

16 participants (11 men and 5 women) volunteered for the experiment. All of them were right-handed. Our experiment started with a question about their prior experience with AR and VR headsets. Twelve participants reported that they *never to rarely* used AR headsets, and the other four answered *sometimes to daily*. Eleven participants answered *never to rarely* for VR headsets, and five of them *sometimes to daily*.

1.4.4 Apparatus

Our experiment ran on a Microsoft HoloLens 1 ($30° \times 17.5°$ screen) and a Leap Motion controller (running Orion 4.0.0+52238). Communication between the Leap Motion and the HoloLens was enabled by a Microsoft Surface Book 2 through WiFi / UDP.

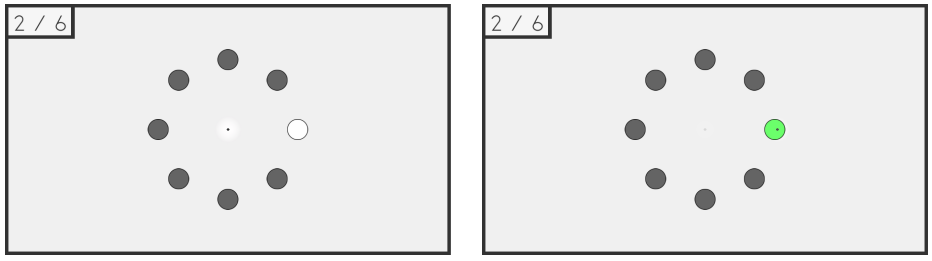
The experiment scene was rendered in an AR vertical window of 47.5×84.5 cm. It was positioned at a distance of 194cm from the headset when starting an experiment session (*i.e.*, its visual size is $24.57° \times 13.96°$). The window then remained fixed in space if users moved their head. The control display ratio for *ARPads* in this specific case is thus 1cm:4.62cm. A 1cm hand movement on an *ARPad* makes the cursor move 1.38° in visual angle, which is 4.62cm in the window (Figure 1.4).

A Leap Motion controller has some limitations regarding the range within which it can accurately track users' hands. In particular, the hand must not be too close to the device (*i.e.*, distance should be at least 8cm). We made users aware of this limit by providing some feedback. When users' hand is getting close to the limit, the border of the AR window progressively turns red, indicating that they should adjust their hand position. The operator instructed the participants about how to interpret this visual feedback at the beginning of the experiment.[2]

1.4.5 Tasks

As in several recent studies (*e.g.*, [52, 101, 141]), our tasks focus on 2D interaction. This does not limit *ARPads* to 2D interactions, however. An *ARPad* can also be used for interacting with 3D content same as mouse input can control 3D software on desktop workstations.

[2] Such "Out of range" events occurred rarely in our experiment. For Indirect Techniques, it happened: (i) in 3% of pointing trials; (ii) between 0.05% and 1.8% of total time per pursuit trial.



(a) (b)

Fig. 1.5: The pointing task. (a) The target to acquire is colored white. (b) It turns green when the cursor enters it. Users have to keep the cursor inside it for 500ms to make it disappear.

Participants had to perform two types of task with each technique. The first type of task aims at operationalizing discrete interactions. It consists in a typical pointing task implemented according to the ISO 9241-9 standard [82]. The second type of task aims at rather operationalizing continuous interactions that typically occur during direct manipulation of objects or during gesture-based input. Inspired by the study conducted to evaluate OwnerShift [52], we operationalize such continuous tasks with the pursuit tracking task introduced by Poulton [165].

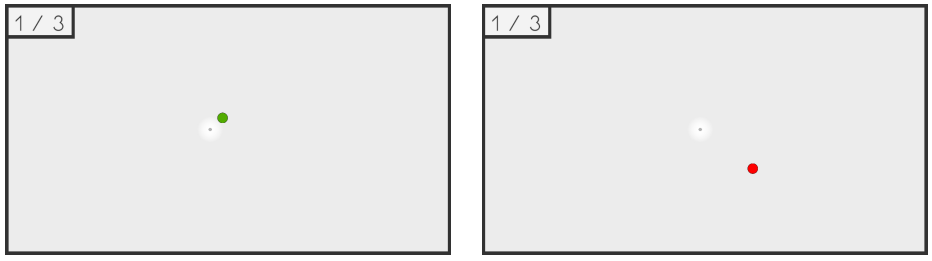
Task 1: Discrete, point-based interactions

Participants perform pointing tasks in diverse directions by acquiring a series of targets that are laid out in a circular manner. As illustrated in Figure 1.5, participants acquire eight targets successively. The next target to acquire is colored white. It turns green as soon as the cursor enters it. The participant then has to keep the cursor inside the target for 500ms (*dwell*) to make it disappear. This makes the next target to acquire turn white. We chose to rely on dwell actions for target selection so as not to involve any additional input device or gesture recognition engine that could have introduced some noise.

All targets in a series have the same size. Some series feature *Large* targets (2.36°), *Medium* targets (1.18°) or *Small* targets (0.59°). Targets are laid on a 7.78°-diameter circle.[3] The difficulty of the pointing task is thus an inverse function of target size.

For each elementary pointing task, we measure acquisition time (*i.e.*, the time interval between the last target's disappearance and the time at which the current target is selected), as well as the number of errors (*i.e.*, how many times the cursor leaves the target before it gets selected). As the cursor is located at the center of the scene at the beginning of a series, its distance to the first target is smaller than the diameter of the ring on which targets are laid out. We thus ignore the first target acquisition of all series.

[3] One visual degree corresponds to 3.39cm in the AR vertical window displayed at 194 *cm* from the headset, and to 1.73cm in motor space for indirect techniques. Our conditions thus correspond to 8 *cm*-, 4 *cm*- or 2 *cm*-targets laid out on a 26.5 *cm*-diameter circle in AR window, and to 1.73 *cm*-, 0.87 *cm*- or 0.43 *cm*-targets on a 5.74 *cm*-diameter circle in motor space for indirect techniques.



(a) (b)

Fig. 1.6: The pursuit task. (a) The cursor is green when it is close to the target. (b) As it is getting further from the target, it turns reddish.

Task 2: Continuous, trajectory-based interactions

Participants have to follow a circular target (0.59° in diameter) that moves in a quasi-random manner within a square of 11.66°×11.66° (Figure 1.6). Participants are instructed to follow this target in order to keep the cursor as close as possible to it. We limited the duration of pursuit tasks to 30 seconds. The target trajectory seemed random to participants, but was actually pre-computed in order to balance the difficulty across participants and techniques. In our experiment design, users had to perform a series of three pursuit tasks with each of the eight techniques. We thus generated three target paths in advance using a sum of four sinusoids for both the x- and -y directions with parameter values reported in [52]. We used the same three paths for each participant × technique block.

For each pursuit task, we consider the mean angular distance between the target and the cursor (we incrementally compute this value at each frame).

1.4.6 Design and Procedure

We follow a within-subject design with the primary factor TECHNIQUE. Figure 1.3 gives an overview of the eight TECHNIQUE conditions. Each indirect technique (*i.e.*, *ARPad*) is a combination of the two secondary factors BODYPOSITION and PADORIENTATION, which correspond to the respective dimensions in our design space. For the two direct techniques, we consider the value of BODYPOSITION to be *Head* but the other secondary factor rather corresponds to the modality: *Gaze* or *Hand*.

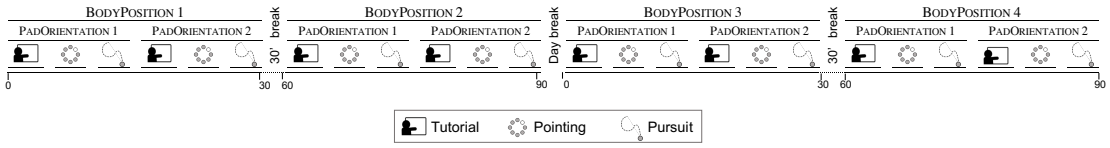


Fig. 1.7: Experiment timeline

Figure 1.7 illustrates the presentation order of trials in our experiment. Each participant completed four blocks, one per BODYPOSITION. The presentation order for these blocks was counterbalanced with a Latin Square. There were two subblocks per BODYPOSITION block, one per PADORIENTATION (for BODYPOSITION=*Head*, there were one *Gaze* and one *Hand* subblock). Presentation order between the couple of subblocks was also counterbalanced across participants and body positions. Finally, within each subblock, the presentation order of tasks was always the same: a tutorial session, followed by all pointing tasks and then all pursuit tasks.

In the tutorial session of a block, the operator quickly introduced the technique to be tested. When the technique was an *ARPad*, the operator used a cardboard sheet to illustrate where the pad stood and what its orientation was. Participants then had to point at eight targets, one at each corner of the screen and one at the middle of each side. If needed, participants were free to perform this training session again at will, before actually starting the block. Participants then completed 2×3 pointing series (one series actually consists of eight elementary pointing tasks as detailed above). Pointing series were grouped by three, which were always presented in an increasing order of difficulty: *Large*, *Medium*, and *Small* targets. The first group of three is considered practice, and the other one is used for analysis. Finally, participants performed three pursuit tasks as described above. Here, the first one is considered practice while the other two are for analysis.

We took particular care not to have a transfer of fatigue from one technique to the other. As Figure 1.7 illustrates, our experiment was divided into two sessions of 1h30mn each, performed on consecutive days. Each session was also divided into two sub-sessions of 30 minutes, separated by a 30-minute break to let participants rest. They could rest longer if they wanted to. In all cases, they started a BODYPOSITION block only after having self-assessed themselves as not tired. We counterbalanced presentation order for the two PADORIENTATION conditions within a block so as not to have always the same transfer direction from one orientation to the other.

Participants had to answer several questionnaires during the course of the experiment. The first one had to be filled in at the very beginning of the procedure. It consisted of the consent form and two questions about their prior experience with head-mounted displays. Then, at the end of each BODYPOSITION condition, they had to rate the mental demand, the physical demand and the perceived performance of the technique for each of the two pad orientations (using definitions and associated 20-point Likert scales from NASA-TLX). Collecting participants' comparative feedback between techniques is somewhat conflicting with the aim of minimizing transfer of fatigue between conditions. As sessions were run across two separate days, participants might have had difficulties recalling conditions that they tested on the first day. To help participants remember each technique, conditions were illustrated with the pictograms used in Figure 1.3. They were displayed in each instruction message all along the experiment, and were printed in the questionnaires.

1.4.7 Results

We use repeated measures ANOVAs and post-hoc paired t-tests for quantitative analyses (with aggregated data, after having checked that there is no strong violation of normality). We use ART ANOVAs and post-hoc paired Wilcoxon signed-rank tests for qualitative analyses. To decrease the probability of Type I errors, we apply Bonferroni-Holm corrections. Our graphs represent data distribution using notched box plots (notches show the 95% confidence interval for the median). We use R [167] to run our analyses. ANOVAs rely on the `ez` [106] and `ARTool` [92] packages. Post-hoc tests rely on functions `t.test` and `wilcoxsign_test` of the `coin` [78] package, and the `p.adjust` function. Charts rely on the `ggplot2` library [212] for box plots and the `likert` package for Likert graphs.

We first analyze data collected in pointing tasks, then in pursuit tasks. For both tasks, we analyze the effect of TECHNIQUE to get a comparison between the different *ARPads* and with the direct baselines (*i.e.*, *Head* condition). We then analyze the effect of PADORIENTATION and BODYPOSITION for indirect techniques only in order to understand how each of the two dimensions of the design space contributes to their performance. These results help to interpret differences between indirect techniques. Such analyses for indirect techniques only consider a subset of data, as PADORIENTATION is not crossed with BODYPOSITION in the *Head* condition. Our experimental design ensures a balanced number of measures across the different groups that we compare in all cases. However, when we analyze data for indirect techniques only, results might have been slightly impacted by the presentation order as the counterbalancing strategy applies to the whole set of technique conditions. We evaluate this impact as low, as the global Latin Square still ensures a balanced presentation order between pairs of indirect techniques. The only impact might come from the presentation of baseline direct techniques in-between two indirect conditions.

Finally, we analyze participants' feedback through questionnaires.

Pointing Task

Direct and Indirect Techniques (Time). Figure 1.8 shows pointing time (*Time*) for each TECHNIQUE, and Figure 1.9 shows a breakdown of these data per TARGETWIDTH. An ANOVA for model *Time* $\sim$ TECHNIQUE $\times$ TARGETWIDTH reveals a significant effect of TECHNIQUE ($F_{7,105} = 9.02$, $p < 0.001$, $\eta^2_G = 0.16$), and of TARGETWIDTH ($F_{2,30} = 9340$, $p < 0.001$, $\eta^2_G = 0.69$) on *Time*. Moreover, the ANOVA reveals a significant TECHNIQUE $\times$TARGETWIDTH interaction effect ($F_{14,210} = 3.28$, $p < 0.001$, $\eta^2_G = 0.06$).

Unsurprisingly, TARGETWIDTH has a significant effect on *Time*. The larger the target, the easier the task, as reflected by the shorter pointing time.

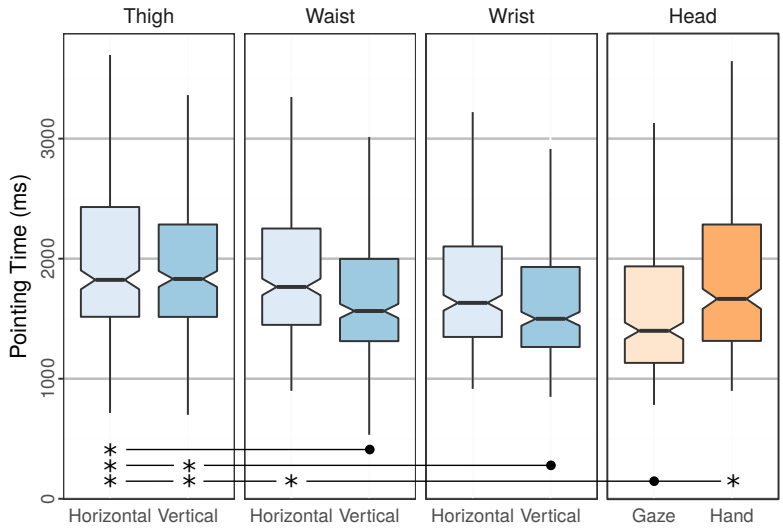


Fig. 1.8: Notched box plot for the pointing time per TECHNIQUE, grouped by BODYPOSITION, then by PADORIENTATION for indirect techniques or modality for direct techniques. Pointing time includes dwell time for selection.

HeadGaze has the best completion *Time* on average. It is significantly faster than the two *Thigh* techniques (p'$s < 0.001$), than *WaistHorizontal* ($p = 0.007$), and than *HeadHand* ($p < 0.001$). We then look at the interaction with TARGETWIDTH. For large and medium targets, we have the same results (with similar p's). However, for small targets, *HeadGaze* is only significantly faster than *ThighHorizontal* ($p = 0.017$) and *HeadHand* ($p = 0.033$). This suggests that *HeadGaze* suffers from an increase in pointing difficulty, which is in line with previous work that discusses the precision issues users face with gaze input [83, 84, 208].

WristVertical ranks second regarding average completion *Time*. *WristVertical* is significantly faster than the two *Thigh* techniques (p'$s = 0.010$). Looking closer at the interaction with TARGETWIDTH, we can see that, for small targets, *WristVertical* becomes significantly faster than *HeadHand* ($p = 0.021$). Here again, we can see that users have relatively more trouble with direct techniques than with indirect techniques for tasks that require more precision. As *HeadHand* is implemented based on a raycast along the headset-hand vector, it might also have suffered from involuntary head movements.

WaistVertical ranks third regarding average completion *Time*. But it is significantly faster than *ThighHorizontal* only ($p = 0.013$). Furthermore, *WaistVertical*'s advantage is for large targets only ($p = 0.008$). Tests do not reveal any significant difference for medium and small.

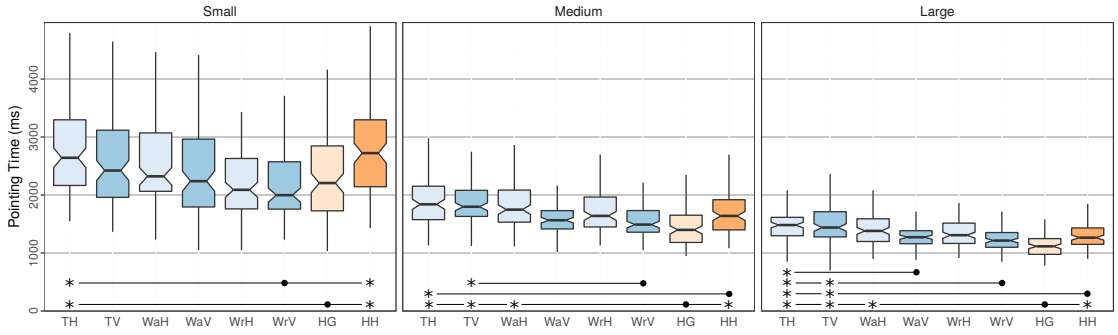


Fig. 1.9: Notched box plots for the pointing time per TECHNIQUE × TARGETWIDTH condition.
TH: *ThighHorizontal*, TV: *ThighVertical*, WaH: *WaistHorizontal*, WaV: *WaistVertical*, WrH: *WristHorizontal*, WrV: *WristVertical*, HG: *HeadGaze*, HH: *HeadHand*

Overall, these results partially support H_2, which states that direct techniques outperform indirect ones. *HeadGaze* actually performs very well, but *HeadHand* does not perform well. However, the *ARPad* that has the best average completion time (*Wrist* × *Vertical*) is not significantly slower than *HeadGaze*. This suggests that indirect input can achieve performance close to that of direct input with gaze.

Indirect Techniques only (Time). Considering data for indirect techniques in isolation, we can analyze the effect of PADORIENTATION with an ANOVA according to model: *Time* ∼ BODYPOSITION × PADORIENTATION × TARGETWIDTH. We observe the same significant effect of TARGETWIDTH ($F_{2,30} = 279$, $p < 0.001$, $\eta^2_G = 0.65$), but we also observe a significant effect of both BODYPOSITION ($F_{2,30} = 12.6$, $p < 0.001$, $\eta^2_G = 0.10$) and PADORIENTATION ($F_{1,15} = 10.1$, $p = 0.003$, $\eta^2_G = 0.04$). We also have one significant interaction effect: BODYPOSITION × TARGETWIDTH ($F_{4,60} = 4.02$, $p = 0.003$, $\eta^2_G = 0.03$).

Our study suggests that participants are significantly faster with *Vertical ARPads* than with *Horizontal* ones (H_3). Moreover, t-tests (corrected $n = 3$) reveal a strict order of performance between the three body positions: *Wrist* outperforms *Waist* ($p = 0.04$), which in turn outperforms *Thigh* ($p = 0.04$). Regarding the BODYPOSITION × TARGETWIDTH interaction effect, t-tests (correction $n = 9$) show that the superiority of *Wrist* over *Waist* is significant for small targets only. But, overall, ***ARPads* seem to perform best when they are positioned at users' wrist (H_4)**.

Direct and Indirect Techniques (Precision). To select a target, participants had to keep the cursor for at least 500ms within the target's boundaries (dwell). We counted a *TargetLeave* (*TL*) event each time the cursor went out of the target, before successfully selecting it. The number of such events provides an indicator of users' precision with the different techniques, as it increases with the difficulty of stabilizing the cursor.

For this analysis, we consider the full set of collected data (direct and indirect techniques). With averages of 1.71 and 1.36 respectively, *HeadGaze* and *HeadHand* have a significantly higher number of TL events than indirect techniques ($p's < 0.001$), which have an average of TL=0.58 by elementary pointing task. Vertical *ARPads* at the *Thigh* position have a slightly higher number of such events (0.74) than other *ARPads*, and actually significantly more than *ARPads* at the *Wrist* location ($p's = 0.013$), which have an average of TL=0.46.

This suggests that **the two direct techniques, *HeadGaze* and *HeadHand*, suffer from precision issues**, which probably explains why their comparative performance decreases with small targets.

Figure 1.10 shows how collected data fit Fitts' law for each technique. We can observe that the point for small targets is systematically above the linear regression line, and even more above the line that would pass through the other two points (large and medium targets). This indicates a "small target effect" [31], with small targets being more difficult to acquire than what Fitts' law predicts. This effect is stronger for the direct techniques than it is for the indirect techniques, and it is particularly strong for *HeadGaze*. This accounts for the low adjusted r^2 for the direct techniques, as well as the high (*i.e.*, bad) slope of their regression lines. Again, these results are consistent with the precision issues that users encounter with direct techniques. Interestingly, we can also notice that *WristHorizontal* has the lowest (*i.e.*, best) slope, but a very high intercept. This suggests that users might have a bit of trouble in initiating a movement with this technique, but that it likely better scales with the task difficulty.

Pursuit Task

For the pursuit task, our measure is the average angular distance between the target and the cursor (*AngularDistance*) over a 30s trial.

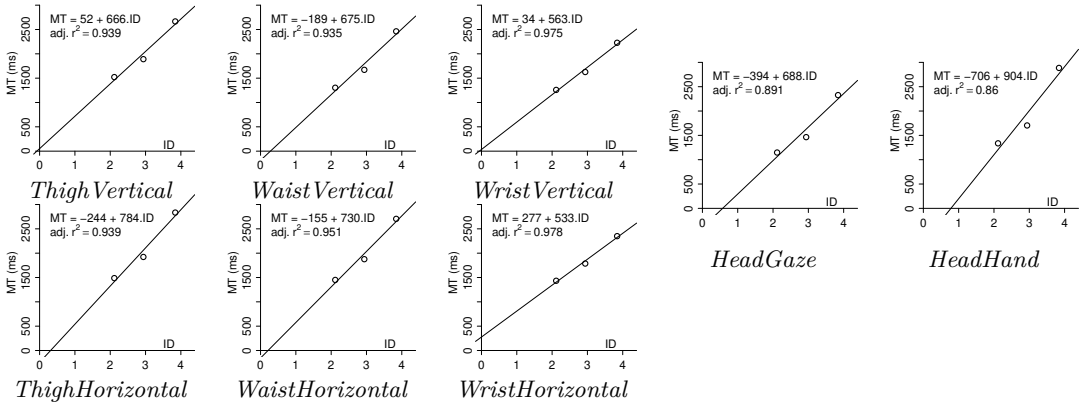


Fig. 1.10: Linear regression analysis with respect to Fitts' law (Movement Time as a function of Index of Difficulty) for all techniques.

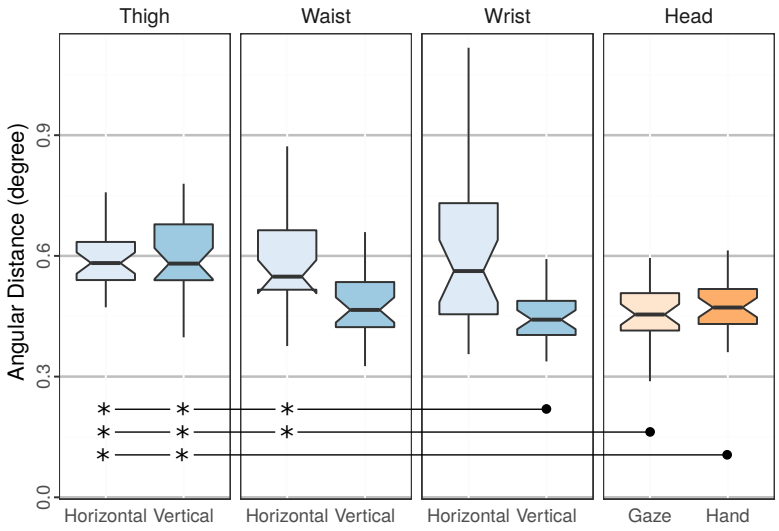


Fig. 1.11: Notched box plot for the average angular distance in *degree* between the cursor and the target during the pursuit. Results are grouped by TECHNIQUE.

Direct and Indirect Techniques (AngularDistance). An ANOVA for the model *AngularDistance* $\sim$ TECHNIQUE reveals a significant effect of TECHNIQUE ($F_{7,105} = 7.65$, $p < 0.001$, $\eta^2_G = 0.27$). Figure 1.11 shows the comparative performance of techniques. **The best performing techniques are *HeadGaze*, *WristVertical*, *HeadHand*, and *WaistVertical*.** Corrected t-tests ($n = 28$) do not reveal any significant difference between these four techniques. This suggests that *HeadHand* performs comparatively better for continuous tasks than it does for pointing tasks.

Pairwise t-tests also reveal a few interesting differences. First, the two *ARPads* at *Thigh* perform significantly worse than the two direct techniques and than *WristVertical* (p'$s < 0.01$). Second, *WaistHorizontal* performs significantly worse than both *WristVertical* ($p = 0.015$) and *HeadGaze* ($p = 0.031$).

Indirect Techniques only (AngularDistance). To analyze the effect of PADORIENTATION, we filter out data for indirect techniques only (as we did for pointing tasks above). An ANOVA for model *AngularDistance* $\sim$ BODYPOSITION $\times$ PADORIENTATION reveals a significant effect of PADORIENTATION ($F_{1,15} = 10.8$, $p = 0.005$, $\eta^2_G = 0.10$) and a significant BODYPOSITION $\times$ PADORIENTATION interaction ($F_{2,30} = 6.63$, $p = 0.004$, $\eta^2_G = 0.09$), but no effect of BODYPOSITION ($F_{2,30} = 1.70$, $p = 0.200$, $\eta^2_G = 0.04$). Pairwise t-tests reveal where the interaction comes from: *Vertical ARPads* perform better than *Horizontal* ones only for *Waist* ($p = 0.033$) and *Wrist* ($p = 0.011$) body positions. Whatever their orientation, ***ARPads* located at the thigh perform particularly poorly for continuous tasks.** But, overall, **our study suggests that *Vertical ARPads* are more efficient than *Horizontal* ones (H_3).**

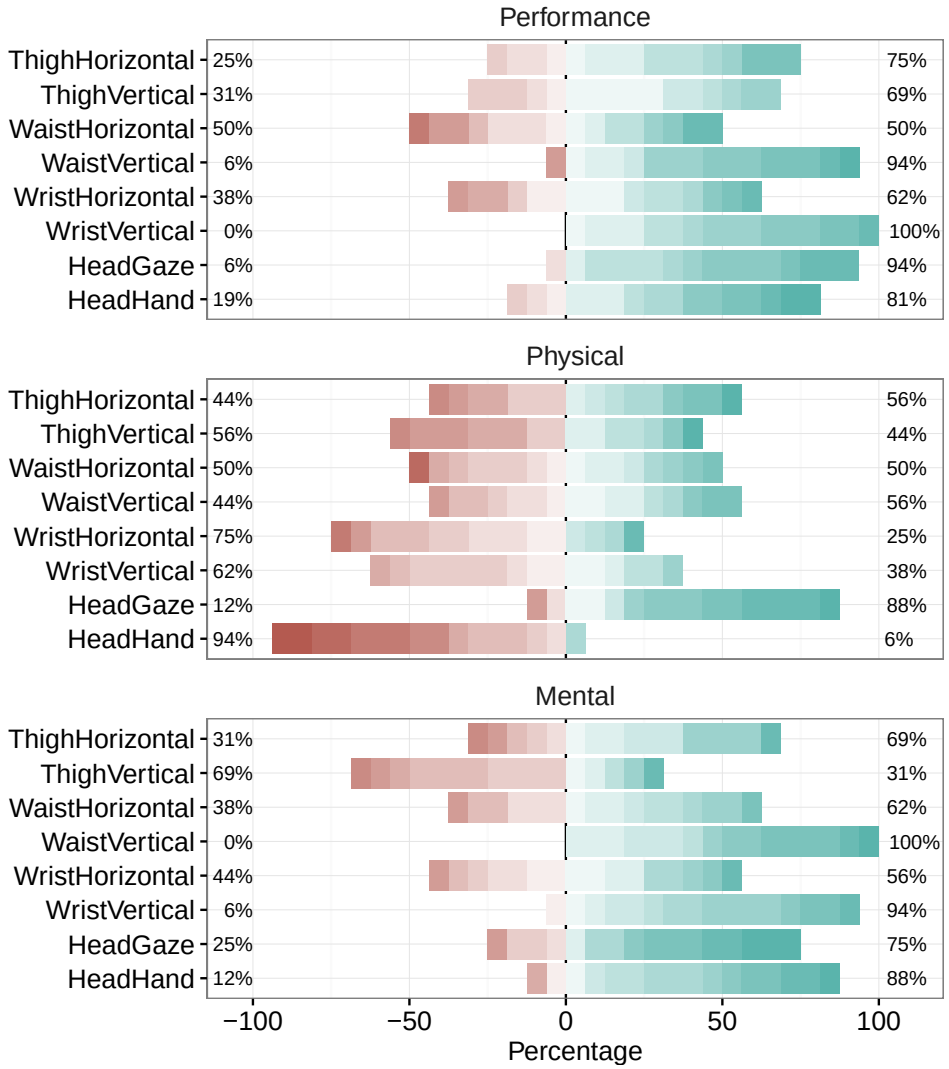


Fig. 1.12: Participants' grades for the following three NASA-TLX scales: perceived performance, physical demand, and mental demand. The plot relies on color coding and binning: each score is assigned a color, and the color patch's size is proportional to the ratio of times this score was given. Right (green) is better.

Qualitative Results

Figure 1.12 shows participants' qualitative feedback for each technique along the following scales: perceived performance, physical demand, and mental demand. ART ANOVAs reveal a significant effect of TECHNIQUE for each of these scales: $F_{7,105} = 4.97$, $p < 0.001$ for perceived performance, $F_{7,105} = 9.70$, $p < 0.001$ for physical demand, and $F_{7,105} = 6.69$, $p < 0.001$ for mental demand.

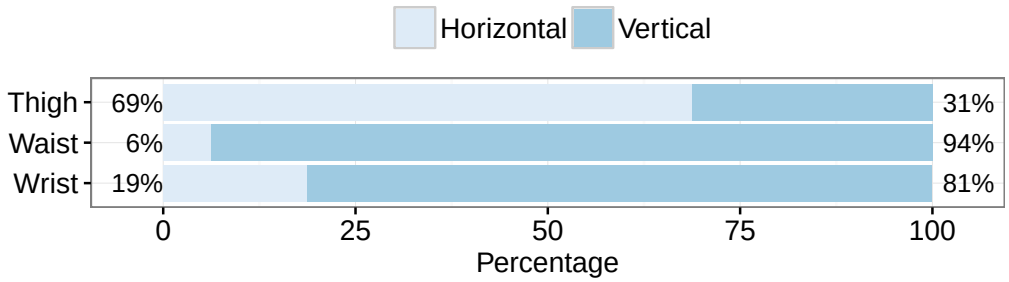


Fig. 1.13: Preferred PadOrientation per BodyPosition for indirect techniques (for direct techniques, all participants consistently preferred *Gaze* over *Hand*).

Participants' perceived performance is consistent with quantitative data. They feel most efficient with *WristVertical*, *WaistVertical* and *HeadGaze*. However, few of the differences are statistically significant. Only *WristVertical* and *HeadGaze* are graded significantly better than *ThighVertical* ($p = 0.040$ and $p = 0.014$).

Grades associated with physical demand are in line with H_1: ***HeadHand* is the most tiring technique on average**. *HeadHand* is actually significantly more demanding than all other techniques (p'$s < 0.038$) except *WristHorizontal* ($p = 0.098$). Regarding physical demand, the only other significant differences are about *HeadGaze*, which is graded as less tiring than *WristHorizontal* ($p = 0.005$) and *WristVertical* ($p = 0.022$). Average grades for *Wrist* techniques are actually high. This is interesting: while participants were fast and precise with *WristVertical*, they also found it quite tiring.

Regarding mental demand, participants found it particularly demanding to interact with *ThighVertical*, which has been graded significantly worse than both direct techniques (p'$s < 0.037$), *WaistVertical* ($p = 0.013$), and *WristVertical* ($p = 0.011$). Also, for *Waist* and *Wrist*, horizontal *ARPads* have been graded significantly more demanding than vertical ones ($p = 0.021$ and $p = 0.030$, respectively).

Regarding PadOrientation, we also asked participants about their preferred orientation for each BodyPosition. Figure 1.13 shows participants' answers. While preferences are unclear between the two *Thigh* techniques ($p = 0.210$), they preferred *Vertical* for both *Waist* ($p < 0.001$) and *Wrist* ($p = 0.021$). We also asked them about their preferences between the two direct techniques: they all preferred *Gaze* over *Hand*.

The results of the above two paragraphs are in line with H_3. The larger the difference between the visual space and the motor space, the less usable an *ARPad* is (more tiring and less liked). However, the *Thigh* seems to be an exception: *ThighVertical* received a very poor grade for mental demand and there is no clear PadOrientation preference between the two *ARPads*. This echoes participants' feedback during the experiment: seven of them mentioned having trouble understanding how the cursor's position was

mapped to hand movements in both *Thigh* conditions. This might be because of *ARPads*' lateral location in comparison with *Wrist* and *Waist* positions which are more centered on the sagittal plane. For example, one participant mentioned that they were expecting vertical *ARPads* at the *Thigh* to be parallel to the sagittal plane (*i.e.*, *Vertical (rotated)* in our design space – Figure 1.1). We designed it this way to be consistent across body positions, but it might be interesting to reconsider our design for the *Thigh* position. For example, we could position the Leap Motion to make it track in front of the thigh as opposed to sideward, as proposed by two participants.

1.5 Implications for Design

Data collected in our experiment support most of our hypotheses, and bring additional insights from which we can derive guidelines about implementing indirect input for AR headsets.

1.5.1 Direct Input vs. Indirect Input

First, users actually perform better on average with direct gaze input than with any of the other techniques we have tested. However, well-designed *ARPads* allow users to reach a similar level of performance overall. In particular, our study showed that direct input suffers from precision issues, and that indirect input might be better suited to interaction with small objects.

Second, as we hypothesized, direct hand input is very tiring. However, our study also showed that direct hand input (*HeadHand*) performs quite well for continuous tasks, better than it does for pointing tasks. It thus remains a good candidate to consider for brief manipulations. However, it quickly causes fatigue, so its use should be limited to manipulations that do not last too long. To give a rough estimate, pursuit tasks in our experiment lasted only 30s but already caused much fatigue. Direct hand input should definitely be considered neither for long manipulations of objects nor for repetitive tasks.

1.5.2 Guidelines for the Design of Indirect Input

The data collected yield guidelines for practitioners regarding *ARPads*'s position as well as their orientation.

Users' performance with *Thigh* was poor in comparison to *Waist* and *Wrist*. Comparison between *Waist* and *Wrist* is more nuanced. Participants performed slightly better with *ARPads* located at *Wrist* for discrete tasks. However, qualitative feedback also showed that they found them more tiring than *ARPads* at *Waist* position. Furthermore, one limitation of the *Wrist* position is that it prevents users from performing bi-manual interactions for, *e.g.*, controlling two cursors. *ARPads* at *Wrist* are good candidates for short-duration interactions, and they might be particularly relevant for some situations such as, *e.g.*, remote control in a seated position. But, overall, *Waist* seems to be the best compromise. Users were efficient and felt comfortable with this body position.

Participants performed better and expressed a clear preference for *Vertical ARPads* over their *Horizontal* counterparts. Despite experience with trackpads on laptops, a difference in orientation between visual and motor spaces seems to have a negative impact.

1.6 Conclusion and Future Work

AR usually relies on bare-hand input. However, this can cause fatigue when performing long-lasting tasks. Relying on device-based techniques is an interesting option that, however, prevents users from keeping their hands free. In this chapter, we answer our first research question by studying *ARPads*, which are mid-air indirect input techniques for AR. We contribute guidelines for designing such indirect input that we derive from an empirical investigation of design space for *ARPads*. Our study compared different designs for *ARPads* and measured their comparable performances with direct input baselines. In our study, gaze input performed very well but suffered from precision issues. Participants' performance with *ARPads* suggest that well-designed indirect input can achieve the same level of performance as direct gaze input. In addition, *ARPads* are particularly relevant for long-lasting or precise interactions as they minimize fatigue and cope with precision issues related to gaze input.

We do not advocate for replacing direct input with indirect input, however. Direct input actually performed well in our study and is probably more intuitive. We rather believe that both types of input should co-exist, allowing users to freely choose the type of input that best suits their context and task. For example, both types of input could be combined in bi-manual techniques where one hand performs direct object selection, and the other hand performs long or fine parameter adjustments using indirect input. Users could also decide to rely on indirect input when they want their interactions to be more subtle [4] or rely on direct input when, on the contrary, they want them to be transparent to others [46]. Supporting the coexistence of both requires implementing mode switching techniques. Such mode switches could be implicit, relying on hand location relative to headset orientation, in the spirit of the Gaze-Shifting technique for pen input [155].

Finally, this preliminary study suggests directions for future work. First, *ARPads* tested in this study implement absolute control with a headset that has a limited-size viewport. Considering absolute control was necessary to test the effect of an *ARPad*'s position on users' performance without any confound from a posture recognition engine (for *e.g.*, clutching actions for relative control). A larger viewport would require considering one or several of the following strategies: increasing the CD gain, enlarging *ARPads*, or implementing a clutching mechanism.

Second, complementing *ARPads* with subtle audio or haptic feedback could help users get a better sense of where the pad stands in the air. This is particularly relevant when relying on technology such as the Leap Motion, which has tracking limitations.

Third, our informal observations suggest that users do not adopt planar movements when interacting with an *ARPad*. A fine-grained analysis of users' movements could help refine the pads' orientation, or even consider pad surfaces that are not planar. Finally, in our experiment, participants were interacting with graphical objects that were displayed in a world-anchored window that was in front of them. It would be interesting to study cases where users are not well positioned in front of the window they are interacting with.

With such interaction, AR could be used for a wider range of daily tasks as it is no longer limited to short-lasting tasks. In this chapter, we have focused on motor performance and comfort when interacting with AR, without reflecting upon its usefulness in a daily context. However, such low-level considerations are a necessary prerequisite, as daily tasks often involve long-lasting operations. Now that we have presented a first investigation over this topic, we can investigate interesting use cases in a daily life context.

2. ENHANCING SMARTPHONE-CENTRIC INTERACTION WITH AUGMENTED REALITY

This chapter is based on the paper "AR-enhanced Widgets for Smartphone-centric Interaction" [19] that has been presented in MobileHCI '21. The co-authors of this paper are Emmanuel Pietriga and Caroline Appert.

2.1 Introduction

Augmented Reality Head-Mounted Displays (ARHMDs) have capabilities that distinguish them from other mobile devices: they render stereoscopic imagery seamlessly superimposed on what users see in their field of view, and they enable manipulation of this imagery via freehand gestures coupled with head- and gaze-tracking. As such, ARHMDs have much potential to complement smartphones and smartwatches, which render imagery on a much smaller physical surface but with a much higher display quality, while also enabling tactile input. Combined together, these different devices are shaping a future where users interact with *Mobile Multi-Device Environments* [62]. In such environments, users can select the best device for a task based on its unique strengths, but they should also be able to use those devices together as one single, powerful, and seamless interactive system.

The combination of ARHMDs with other personal devices has opened up a large design space. Of particular interest is the combination of smartphones and ARHMDs, which can be made to work in tandem, as explored at length in the BISHARE design space [218]. For instance, smartphones can be used as high-precision input devices to manipulate spatial AR content [135], while ARHMDs can enlarge a smartphone's display capacity [60, 148] or show 3D content in the air. These are only two examples from a much richer design space, in which many possibilities remain to be explored. In this chapter, we explore a specific part of the BISHARE design space [218], enhancing smartphones with ARHMDs. The BISHARE design space identifies high-level dimensions of smartphone+AR HMD combinations, covering a very large set of designs but learning out details. We introduce a lower-level design space to specifically describe the case where the AR HMD is used to enhance a smartphone-centric application. Our two main dimensions are (1) the function of the holographic content, and (2) its location around the smartphone's screen. We populate this design space with examples from the literature and novel ideas.

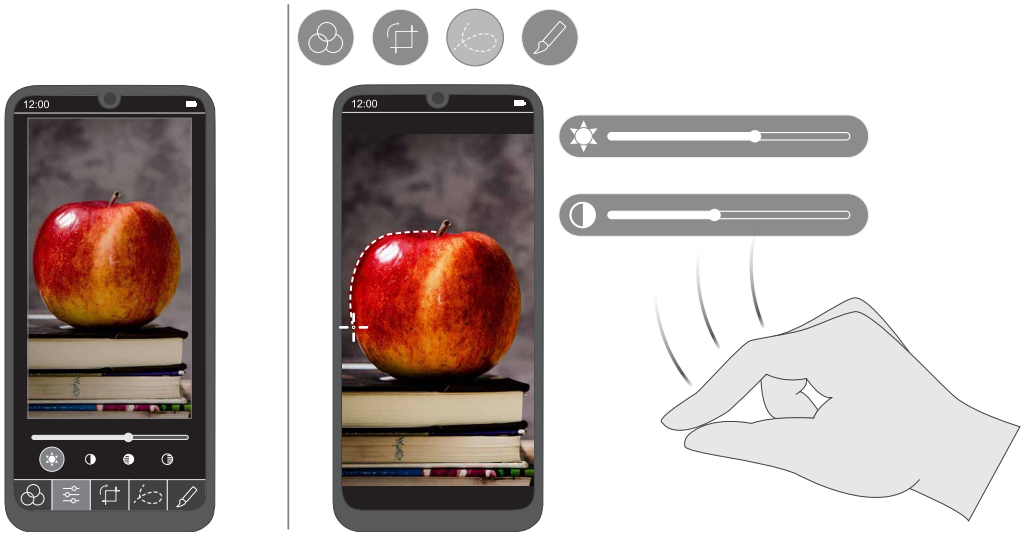


Fig. 2.1: Offloading controls of an image editing application running on a smartphone. Left) UI Widgets take significant screen real-estate when displayed on the phone; and precise selections are difficult with touch input because of finger occlusion. Right) Default UI distribution: 1) *output*: widgets are displayed in the air, freeing screen real-estate on the phone to show more content; and 2) *input*: the pointer can be controlled indirectly with movements performed in the air, causing less occlusion of the image being edited.

Then, we study a suite of AR-enhancements for phone-centric interaction. Acknowledging the strong *legacy bias* associated with smartphones, which makes users resort to well-known interaction styles whenever possible [20, 218], AR-enhanced widgets are augmentations of those traditionally found on this type of device. We make use of AR capabilities to distribute output by offloading widgets "in the air" around the phone,[1] as well as to distribute input between touch and mid-air gestures. Both distribution strategies free up precious screen real-estate on the phone, and limit situations where content gets occluded as well, since users interact partly "around the device" [85, 93]. Additional input degrees of freedom may also help users perform tasks that require precision.

These UI distribution strategies that consist of offloading widgets in the air when applicable, and making indirect control possible through movements in the air (as a complement or replacement to touch). Figure 2.1 illustrates the potential advantages of such an approach: 1) more widgets always visible in the air and more space on the phone's screen to display content in high resolution, and 2) indirect control with hand movements in the air to avoid potential occlusion and to enable higher-precision control. We discuss design considerations about applying such a strategy to phone widgets, and report on a series of exploratory studies to identify viable designs and estimate their performance.

[1] The concept of such widgets actually floating in the air around the phone has been popularized in science fiction shows such as, *e.g.*, *The Expanse*. Querying the Web for `"the expanse hand terminal"` yields artists' renderings that accurately illustrate the idea.

Our contributions are:

1. a design space of smartphone + ARHMDs applications;
2. a series of detailed designs for AR-enhanced phone widgets;
3. an exploratory study of these widgets compared to their state-of-the-art, touch-only counterparts.

2.2 Related Work

Phone-centric distributed UIs constitute only two cells in a larger design space of *bidirectional interactions between smartphones and head-mounted augmented reality* [218], which is itself only one item in the *cross-device interaction taxonomy* [20]. We refer the reader to these two recently-published and complete articles, and only focus here on work that directly relates to distributed UIs in a mobile context: interacting above and around handheld or wearable devices; augmenting mobile 2D input; multi-device mobile systems involving augmented reality technology.

2.2.1 Interacting Above and Around Mobile Surfaces

While direct touch remains a very effective way to interact with a mobile device, bare-hand interaction around the device has several interesting properties. It enables a greater physical input volume beyond the small touchscreen [24, 85, 69, 70], sometimes with higher expressiveness [34, 63]. It partially addresses problems of content occlusion [24, 85, 70, 63]. It provides possibilities to lay out widgets beyond the touchscreen [105], or organize multiple pieces of content relative to the user's body and quickly access them [33, 73].

One of the challenges is to accurately track the fingers relative to the device and to segment gestures. While many systems use external motion tracking systems for prototyping purposes [85, 69, 73], some others actually try to embed the tracking solution in the mobile setup. The early SideSight system [24] used IR proximity sensors mounted on the device, tracking finger gestures performed on the surface upon which the device was laid. Skin Buttons [105] also use IR sensors to detect taps on icons projected from a smartwatch on the surrounding skin. Air+Touch [34] uses a miniature depth camera attached to the phone. Similar to SideSight, GlassHands [63] recognizes surface gestures around the phone, but achieves this without additional sensors, by tracking the reflection of the user's phone and hands in her sunglasses with the phone's front-facing camera. Ether-Toolbars [171] explore a similar approach, but uses a reflective surface mounted above the camera. Tracking can also rely on magnetic sensing. Ketabdar *et al.* [93] detect changes in the magnetic field around the phone produced by the circular or linear movements of a finger ring. Nenya [9] also uses a ring, but detects twisting and sliding gestures, which are mapped to selection and click, respectively. Abracadabra [70] has users wear a magnet below their finger to interact around a smartwatch.

One downside of mid-air around device interaction is fatigue [76]. Jones *et al.* [85] report that participants performed equally well with mid-air and with touch interactions, and that they found mid-air interactions enjoyable but tiring as well. Participants also raised concerns about arm fatigue with AirPanes [73]. This suggests that the use of mid-air gestures should be restricted to occasional, non-lasting interactions that complement touch interactions. Air+Touch [34] explores such an approach, interweaving touch events and short in-air gestures. Empirical data about user performance and fatigue is not reported, unfortunately.

2.2.2 Augmenting Mobile 2D Input

Because we consider AR as a means of enhancing input with mobile devices, our work also relates to mobile interaction techniques that add more input degrees of freedom to trigger commands or adjust parameter values. Grossman *et al.* track a pen in close proximity to a portable screen, letting users invoke so-called Hover Widgets [59] to trigger commands with small 2D gestures performed *in-situ*. The Hover Cursor [149] tracks fingers hovering the phone's screen, in this case to facilitate the selection of small targets by offsetting the pointer, thus avoiding occlusion of the target of interest. Earlier, Shift [206] had explored a complementary approach to the well-known fat-finger problem, offsetting a copy of the area under the finger when in contact with the screen to keep that region visible and enable precise selection.

The above techniques can be seen as augmenting the capabilities of the screen itself. Other techniques have investigated a more indirect way of augmenting 2D widgets, using the mobile device's side buttons. SidePress [191] uses two buttons on one of the phone's sides, that can sense different pressure levels. The buttons can be used for any bidirectional selection and navigation task. Going further, the Power-up Button [190] combines pressure and proximity sensors to let users control 2D widgets with the thumb holding the device.

2.2.3 Multi-device Mobile Systems using AR

AR-enhanced phone-centric interactions are part of the broader category of Distributed User Interfaces in which the system is aware of the relative spatial location of all devices involved: smartphones, tablets, and combinations thereof [168, 118, 61]. These often focus on use cases where one or more users temporarily combine several devices in a stationary context, typically laying them out on a table. Gluey [180] is a conceptual interactive system where devices, which can be in different locations, are combined thanks to an ARHMD, using the latter for tracking and as a means to transfer content between devices.

ARHMDs also enable distributed user interfaces adapted to a mobile context. As mentioned earlier, ARHMDs and mobile touchscreens have very different capabilities, that can complement one another effectively when made to work in tandem, as observed empirically

with MultiFi [60]. The BISHARE design space [218] organizes smartphone+ARHMD combinations into two broad categories: *Phone-centric* and *HMD-centric*. The latter category covers cases where the handheld device enhances spatial interaction with AR content. For instance, the handheld device becomes an alternative or a complement to bare-hand input for AR/VR content manipulation [23, 196] and navigation [126], with applications to, *e.g.*, CAD [130] and gaming [135]. The former category rather includes cases where the HMD is used to enhance interaction with the handheld device. The AR-enhanced interactions we focus on fall under this category, and more specifically in subcategories *phone-centric distributed UI* (D2P) and *phone-centric distributed input* (D1P).

Such phone-centric combinations, in which the ARHMD is mostly used to add more input and output capabilities to the phone, have gained attention in the HCI literature. VESAD [148] is an example of augmenting a phone's visual output. The technique virtually extends the phone's screen by displaying additional 2D content in AR around the phone, keeping that content spatially aligned in the same plane. A VESAD proved more efficient than a phone only on a manual classification task. However, Eiberger *et al.* [44] report that the combination of a physical display with AR is not always beneficial. In their study about the integration of information from different depth layers, participants had to perform a visual search task using a combination of HMD and smartwatch, or using an HMD only. Their performance was significantly lower in the HMD+smartwatch case, in terms of both speed and error.

In the above examples, the AR imagery consists of an expansion of what is shown on the mobile device. But AR can also show complementary information of a different nature, as illustrated by WatchThru [211] and mobile true-3D displays [182]. MultiFi [60] goes one step beyond, making interactive widgets run across devices and ARHMD, adapting both their input and output based on the respective fidelity of devices involved: screen resolution and physical size, direct *vs.* indirect input. For instance, the smartphone's high-resolution screen can be used as the primary display for an item in focus, while the ARHMD will provide lower-resolution information about surrounding items in the same list. Or the phone's touchscreen can be turned into a fullscreen keyboard, the edited text being shown in mid-air. MultiFi essentially focuses on how to make widgets span devices with different input/output characteristics. Interaction remains limited to basic touch and spatial pointing. Finally, VESAD can also be used to show additional windows or widgets around the phone. The authors briefly suggest this possibility [148], but report neither on the prototyping nor the evaluation of such AR-based enhancement techniques.

Extending interactive surfaces to the air can involve input (*e.g.*, mid-air movements), output (*e.g.*, AR imagery) or both – though they have so far been mostly studied separately. Such strategies are promising research avenues. However, the actual, quantifiable pros and cons of applying these extension strategies to distribute UI phone controls in an actual ARHMD+phone set-up have not been explored. In this chapter, we contribute a design

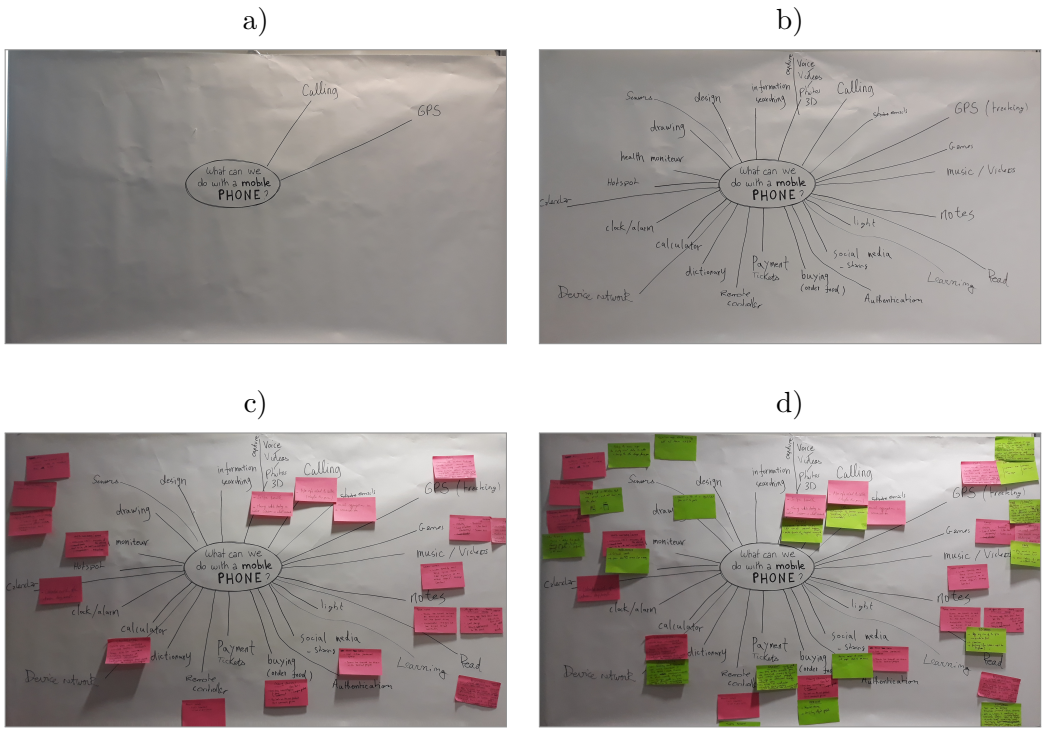


Fig. 2.2: Evolution of the paperboard during the brainstorming session: a) the moderator introduces the first question Q_1 and gives a couple of answers, b) the moderator populates with answers from participants. c) the moderator populates with problems raised with Q_2, d) the moderator populates with solutions proposed with Q_3.

space to help interface developers explore the set of AR-enhancements to mobile devices. Then, we discuss some specific phone-centric widgets' UI design, implementation and performance aspects individually and, where possible, in combination with a representative set of phone controls.

2.3 Workshop

As a first step into the investigation of AR-enhanced phone interactions, we adopted a user-centered design approach with a workshop in order to observe how users would like AR to enhance their interactions with their mobile devices.

We organized a workshop with 13 people (4 females, 9 males) aged from 20 to 33. All of them were familiar with mobile devices, 11 of them had knowledge about Human-Computer Interaction, 3 had already worked with Augmented Reality. Participants were split into 4 groups of 3 to 4 people. The groups remained the same for the whole workshop.

The workshop lasted 2 hours and was organized into 2 parts. The first part was a brainstorming session about what phone applications or use cases would take advantage of AR enhancements. The second part was a storyboarding session where participants had to create a storyboard to detail one of the applications/use cases they had proposed during the brainstorming session.

2.3.1 Brainstorming

The brainstorming session lasted 45 minutes. The moderator asked the three following questions, one after the other:

Q_1 What can we do with a mobile phone?

Q_2 What are the problems users can face in such cases?

Q_3 What AR-based solutions can we think of to address such problems?

Figure 2.2 illustrates how the paperboard evolved during the whole brainstorming session. After each question, participants had 15 minutes to work in groups and provide answers. The moderator populated the paperboard with answers given from participants. Answers to Q_1 were written down in **black**. They are the varying use cases participants could think of. Problems identified when answering Q_2 were marked down on **pink** post-it notes close to their related use case. Finally, solutions coming from Q_3 were reported on **green** post-it notes that were positioned close to the problem they address. The moderator explicitly said to the participants that they could either build on previous answers or propose novel elements if they wanted to.

Q_1 What can we do with a mobile phone?

Altogether, participants grouped their ideas into categories that we can summarize as follows: (1) *Media consumption*, (2) *Localization*, (3) *Gaming*, (4) *Security*, (5) *Social Network*, (6) *Sport*, (7) *Shopping*, (8) *Information*, (9) *Network Management*, (10) *Communication*, (11) *Time*, (12) *Tools*, (13) *Design*, (14) *Wallet*, (15) *General*.

Q_2 What are the problems users can face in such cases?

To think about issues, participants were allowed to use their phones during the workshop. The moderator also encouraged them to reflect on specific issues that they personally faced in the past (critical incidents). Some of the issues they raised are related to hardware limitations while others are more related to UI design, for which AR could provide some solutions.

They identified many issues about applications that relate to physical limitations. (1) screens of smartphones are too small for many applications. For example, when reading

a book, users often zoom in and out. (2) Screen luminosity can have a detrimental effect on users' eyes. (3) The processor speed is often limited. (4) There is a lack of physical buttons, and software controls on the screen consume a lot of valuable display real estate. (5) The battery duration is sometimes short. (6) Memory is limited. (7) There is a lack of haptic feedback due to the flat screen. (8) Sensors are often inaccurate, which can be a problem while doing *e.g.* health monitoring. (9) Internet connection can be poor.

Some other issues more relate to users' perception and psychology. (10) Showing some content on the phone to another user means getting physically close to that user, and potentially invading their personal space. (11) When objects such as books are digitalized, users might miss the physical feeling of it. (12) Smartphones are addictive. (13) Notifications while focusing on a task can be disturbing. (14) As the smartphone is a mobile device, users might get disturbed by the external environment. (15) Representation of the real world through the smartphone may be misleading, for example, when a user places an order, the product's physical features might be actually different from what the user has in mind. (16) Phones typically accommodate a single window, which makes multitasking difficult.

Some of these findings echo mobile devices' issues described by the Nielsen Norman group. In particular, one of their reports [146] lifts some of the issues that were also mentioned by our participants. It also mentions one more which is (17) the inaccuracy of touch interaction. This issue is described as resulting from, first, the small size of the screen, which causes a rise of the interaction's cost and makes the user relies on short-term memory and, second, interruptibility which is inherent to those devices and leads to a fragmentation of users' attention. Both of these causes have been explicitly proposed by our participants.

Q_3 What AR-based solution can we think of to address such problems?

Participants had to answer this question by giving concrete examples about how they would solve the listed issues using AR. There was only one rule imposed: the phone should remain the primary device and should not be turned into a control device for an AR-only application. Following this principle, users imagined several AR-based solutions that can be categorized by action. One of the most cited ideas during the workshop session was (1) *to relate the content displayed on the phone to the physical world*. Participants mentioned, for example, that they would like to see the physical aspect of a digital book when reading it. They also proposed the use of AR to (2) *enlarge* the phone's screen; (3) *add interaction*, either to free the phone's screen from large buttons or keyboards, or to add an new layer of interaction; and (4) *add content*. They proposed this content to be either *independent* from the phone's content (*e.g.* notifications) or *dependent* from it (*e.g.* an additional information layer). Being able to (5) *abstract* the data displayed on the phone was also perceived as useful by the participants. In particular, such abstraction can give a new perspective of the content (*e.g.* an overview of a map or an outline of a text document). Finally, they discussed having multiple views of the phone's screen,

either for sharing one view with another user or to avoid uncomfortable postures that result from the fact that the phone's screen is colocated with their hand. Indeed, users either have to look down toward their hand or maintain their arm in an upper posture. One solution can be to (6) *duplicate* the content of the screen and move the screen's copy to a convenient location.

2.3.2 StoryBoarding

During this part, participants had to select and illustrate one of the given AR-based solutions that they had proposed or that has been proposed by another group. We asked participants to provide two storyboards per group. The first one should be about the issue they decided to focus on. The second one should present the AR-based solution, showing how AR could be useful in this situation. Figure 2.3 is an example of an illustration from participants.

Fig. 2.3: Example of a storyboard created during the workshop: a) description of the issue encountered with a mobile phone, and b) illustration of a solution using an AR Head-Mounted Display

2.4 Design Space

The workshop reported above revealed that there is a large range of possibilities for combining an AR headset and a mobile phone. Analyzing solutions proposed by our participants, we created a design space for AR enhancements to mobile phones. We focused on how AR could complement mobile devices, excluding the cases where mobile devices are used as a controller to manipulate AR content, which is an approach that has already been explored by others [109, 130, 213]. Our design space, which is illustrated in Figure 2.4, features two main dimensions: (1) the *location* of the holographic content, and (2) its *function*. We populated it with both projects and ideas that we found in

the HCI literature (in **bold**) to demonstrate its descriptive power and then used it to generate ideas (in *italic*). Some of these ideas led to actual prototypes that we describe in section 2.4.3.

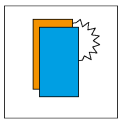
2.4.1 Dimensions of the Design Space

In this section, we detail all the dimensions of our design space. For each dimension, we give a textual description as well as a pictogram in which **orange** represents the holographic content and **blue** represents the mobile device.

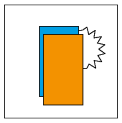
*Dimension **LOCATION***

This first dimension describes the location of the holographic content. It can be relative to: (1) *the handheld device*, (2) *the user*, or (3) *the environment*.

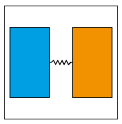
Relative to the handheld device The holographic content is anchored to the handheld device. It can have the same orientation than that of the device's screen (*i.e.*, *coplanar*) or not (*i.e.*, *not coplanar*). When **LOCATION** is *coplanar*, we distinguish cases depending on whether the holographic content is *above*, *behind*, or on the *side* or the device.



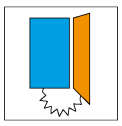
Behind (Coplanar) The holographic content is located behind the handheld device. The user can see only one of the two contents at a time.



Above (Coplanar) The holographic content is superimposed on the handheld device content. It can have a varying level of transparency.

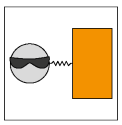


Side (Coplanar) The holographic content is located somewhere around the handheld device.



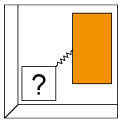
Not coplanar The holographic content is displayed with a different orientation than that of the handheld device's screen.

Relative to the user The holographic content is anchored to the user, meaning that spatial manipulations of the handheld device do not affect the location of the holographic content.

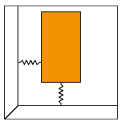


Relative to the user The holographic content is anchored to the user. Its location is defined as a function of the user's location.

Relative to the environment The holographic content is spatially registered in the environment. It can be *anchored* to an object or it can be *floating* in the environment at a fixed location.



Anchored The holographic content is anchored to a specific object in the environment. Manipulating that object will have an impact on the location of the holographic content.

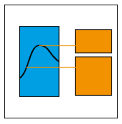


Floating The location of the holographic content is defined as a 3D position in the world. It appears as floating in the air at a fixed position.

Dimension ***FUNCTION***

This second dimension focuses on the function of the holographic screen relative to the mobile device. The holographic layer can be used to display additional data, to provide users with more powerful ways of interacting with the content displayed on the handheld device, or to enhance the presentation of this content.

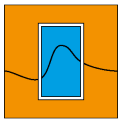
Add data The holographic view contains data that is not on the handheld device. Such data can either be: *Dependent* or *Independent* from the handheld device's content.



Dependent The holographic screen displays data that relate to what is displayed on the handheld device's screen. In that case, the holographic content is dynamically updated as the content displayed on the handheld device changes or as the user navigates it.

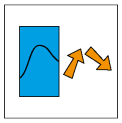
Independent The holographic screen displays data that are not related to the data displayed on the handheld device's screen.

Enhance presentation AR can be used to enhance the presentation of the content displayed on the handheld device, without providing additional data.

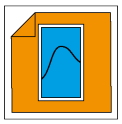


Enlarge The holographic content is used to provide a larger viewport over the handheld device's content.

Duplicate A copy of the handheld device's screen is displayed in AR.

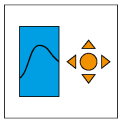


Abstract The holographic screen presents the same content as the handheld device but with a different level of abstraction (*e.g.*, an outline of the content displayed on the handheld device).



Relate to physical world The holographic layer is used to display virtual cues that make the metaphor with the physical world stronger.

Add interaction AR can also provide a more diverse range of controls to interact with the handheld device.



Add interaction The holographic screen features widgets and controllers to manipulate the content displayed on the handheld device.

2.4.2 *Populating the Design Space with Existing Solutions*

To populate this design space [2], we first sorted out AR-enhanced solutions that we could find in the HCI literature. We only considered papers that present a handheld device-centric application. Indeed numerous papers rather present AR applications that use a smartphone or a tablet as a tool for interacting.

Using AR to enlarge a limited device's screen is a concept that has been explored by Feiner and Shamash in early 1991. They describe *Hybrid User Interfaces* [51], proposing to enlarge a computer's screen with AR content. This allows users to keep track of windows that are outside the screen (*Enlarge* × *Side*). Feiner also presented the *Touring Machine* [50] in 1997. This project links anchored AR content related to their university's buildings with their description written on a tablet (*Dependent* × *Anchored*). More recently, researchers have resumed investigating a possible synergy between mobile devices and AR. For example, Budhiraja et Al. [22] organizes the different ways of combining a smartphone with an AR system into the following three categories: a user can either interact with the phone only, use the phone as input for AR, or use the phone to show additional data related to the AR scene. This latter category corresponds to the phone-centric case we are interested in and falls within the *Dependent* × *Anchored* cell of our design space. Wenig *et al.* presented WatchThru [211], an application that displays AR content above a smartwatch by adding a piece of plexiglass on top of the watch. Authors illustrate their concept with *e.g.* a GPS application where the AR screen is used to show the next direction to the user. This application belongs

[2] When creating this design space, we noticed that some dimensions' values could not be crossed. In particular *Duplicate* ×*Front* cell. This would mean superimposing two identical layers, which makes no sense. This cell is colored in **black** in our design space.

FUNCTION	LOCATION: Relative to the digital Device — Coplanar — Front	Coplanar — Side	Coplanar — Back	Not Coplanar	Relative to the user	Relative to the environment — Floating	Relative to the environment — Anchored
Add content — Independent		**VESAD** [148] **BISHARE** [218]		**VESAD** [148]	**Personal Cockpit** [48]		
Add content — Dependent		**VESAD** [148] **DesignAR** [173] **Multi-Fi** [60] *Task Selection* **BISHARE** [218]		**BISHARE** [218]		*Locate photos in space*	**Touring Machine** [50] **Interaction techniques for HMD-HHD** [22]
Enlarge	**DesignAR** [173]	**VESAD** [148] **Multi-Fi** [60] **Hybrid User Interfaces** [51] *Ambilight*					
Duplicate			*Back of device mirror*		*Ergonomic view*		
Abstract			*Context sharing*	**WatchThru** [211]			
Relate to Physical World							*Contextual electricity map*
Make more interactive	**DesignAR** [173]	**VESAD** [148] **BISHARE** [218]				**Multi-Fi** [60]	*Virtual joystick*

Fig. 2.4: Our design space for combining Augmented Reality with mobile devices. The first dimension corresponds to the position of the holographic content. The second dimension represents the function of the AR screen with respect to the mobile device.

to the *Abstract* × *Not Coplanar* cell of our design space. With *Multi-Fi* [60], users can *e.g.* navigate in a menu by moving their phone close to an element in the 3D environment, or navigate a map displayed on a smartwatch, having the context in AR (*Add interaction* × *Floating*). In the VESAD project [148], Normand and McGuffin present concept applications that use the space around the phone to display more content. For example, they propose to provide AR tooltips around the smartphone's screen widgets (*Dependent* × *Side*). They also study different possibilities for enlarging the smartphone's screen (*Enlarge* × *Side*). more recently, BISHARE [218] organized a wider design space of bidirectional interactions between AR and smartphones. While this design space covers a larger range of applications including the use of smartphones as a tool for AR-only applications, some cells fit within our design space. For example, they suggest adding AR controls or independent data around the smartphone which corresponds to our design space's respective cells: *Add interaction* × *Side* and *Independent* × *Side*. DesignAR [173] is another tool for combining a tablet and an AR HMD. More than providing the holographic screen with more controls (*Add interaction* × *Side*), DesignAR [173] proposes to enlarge the tablet's screen to make data accessible in 3D and then, falls within the scope of *Enlarge* × *Front*. Finally, techniques such as the one presented in Personal Cockpit [48] could be use in association with mobile devices, showing some independent content to the user (*Independent* × *Relative to the user*).

2.4.3 Generating novel solutions

We can observe that some cells have not been explored yet, opening spaces for creativity. In this section, we present novel solutions that our design space inspired us. To generate these solutions, we came back to some problems that participants raised during the workshop. We then systematically considered the cells of our design space to think of possible ways of address these issues. We came up with a set of solutions that are displayed in italic in Figure 2.4. Some of these solutions are still conceptual while others have led to actual prototypes that we developed.

Concept applications

Locate photos in space This application would display photos from the phone's gallery at the exact position where they have been taken from. Here, the holographic content is dependent on the smartphone's one, and photos are floating in the 3D space (*Dependent* × *Floating*). Figure 2.5 illustrates this concept application.

Ergonomic view This application would allow users to get a view of their current screen in front of their eyes while holding a good posture for the neck, preventing them from holding their head down. The holographic screen is a copy of the actual phone's screen, located along with the user's sight (*Duplicate* × *Relative to the user*). We illustrate this concept application in Figure 2.6.

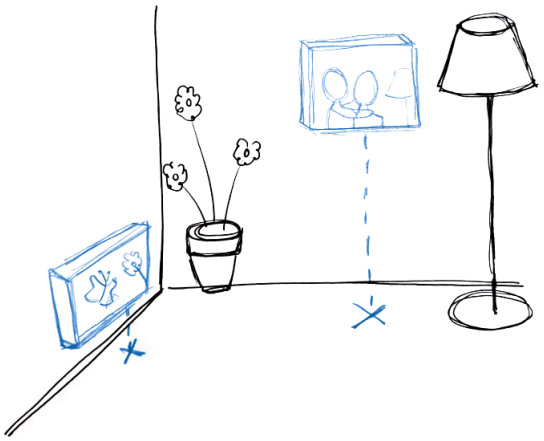


Fig. 2.5: Rough sketch representing the *Locate photos in space* application. Photos are displayed where they were taken from.

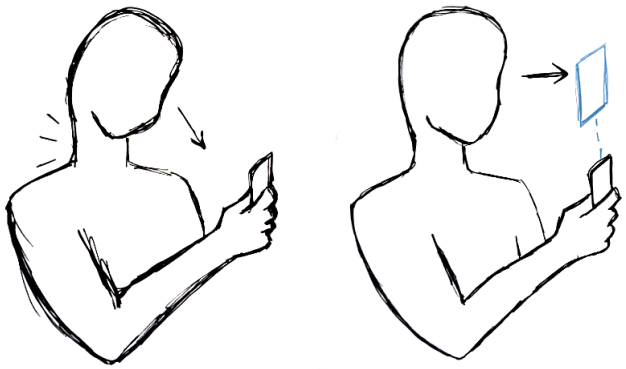


Fig. 2.6: Rough sketch representing the *Ergonomic view* application. A copy of the smartphone's screen is displayed in front of the user's eyes. The left image represent the user looking at their smartphone while holding a bad posture for their neck. The right image represent the same user using the *Ergonomic view* application, holding a good posture for their neck.

Context sharing This application would allow others to quickly get what the user is doing by glancing at the back of their phone. A context icon corresponding to the task the user is doing would be displayed behind the phone (*e.g.*, musical note when listening to music, envelope when reading messages) (*Abstract* × *Behind*). Without revealing the actual content on the handheld device, this application would give a bit of context to others so they know whether they can interrupt the smartphone's user or not. We illustrate this idea in Figure 2.7.

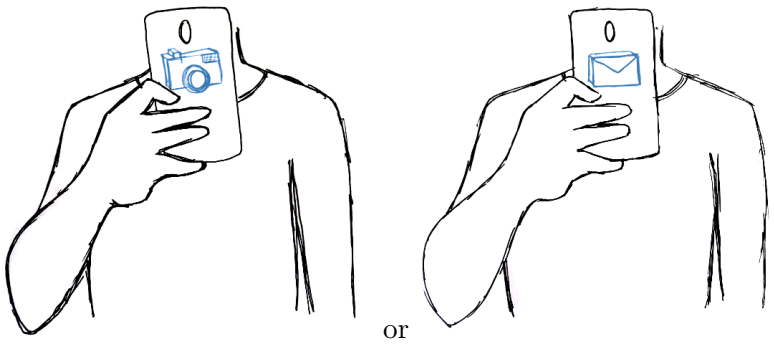


Fig. 2.7: Rough sketch representing the *Context sharing* application. A holographic icon is displayed behind the smartphone to indicate what task the user in currently doing. On the left, the user is taking pictures using the camera. On the right, the same user is reading emails.

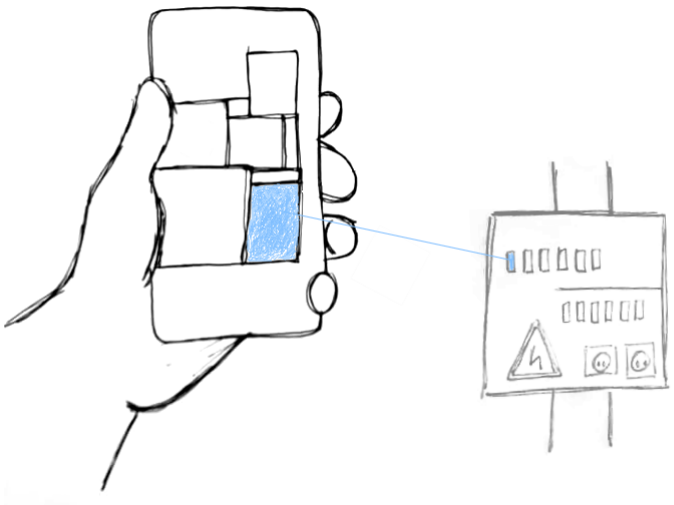
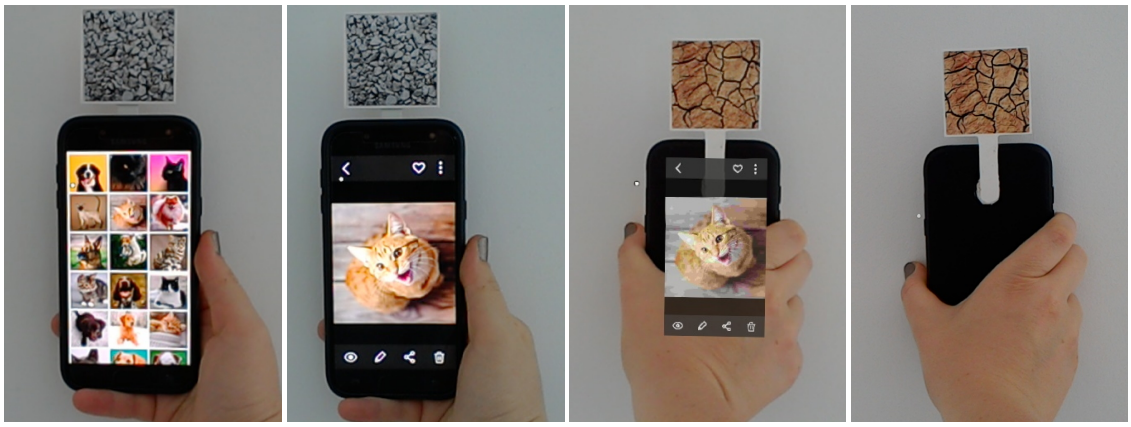


Fig. 2.8: Rough sketch representing the *Contextual electric switchboard* application. The user selects one room of a map displayed on their smartphone. A holographic link identifies which switch of the electrical panel relates to that room.

Contextual electric switchboard This application would display the floor plan of a house. When in front of heir electrical board, a user could select a room and get an AR link that relates that room to a specific switch (*Relate to physical world* × *Anchored*). Figure 2.8 illustrates this concept application.



a) b) c) d)

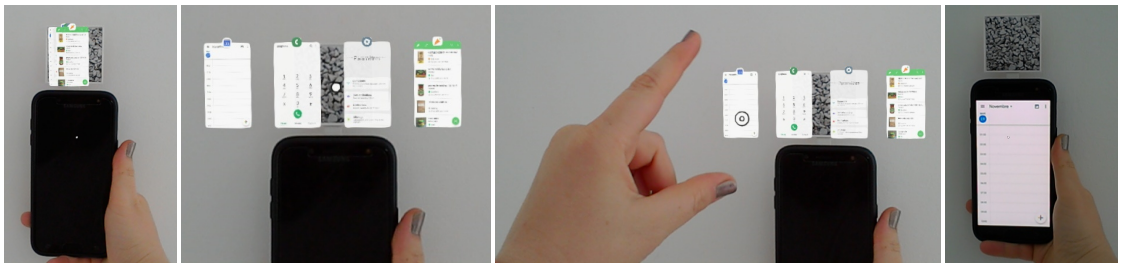
Fig. 2.9: Interaction steps with the *Back of device mirror* prototype: a) the user navigates the smartphone's gallery; b) the user selects a picture in the smartphone's gallery; c) the user shows the picture to their friend, keeping track of what their friend sees thanks to the duplicated view behind the smartphone; d) the duplicated view disappears, letting the user know that the smartphone went into sleeping mode.

Prototypes

We also implemented actual prototypes for some ideas that we considered particularly promising. The first three prototypes were developed with a Samsung Galaxy J5 combined with a Microsoft HoloLens 1. As for software, we used Unity 2018.4.9f1 and Vuforia 8.3.8. The smartphone was equipped with two Vuforia markers on top, one with the same orientation as the smartphone's screen and one on its back. In this section, we also present two projects that have been developed by interns that we supervised. Both have been coded using a Microsoft HoloLens 2 and a Samsung Galaxy A50.

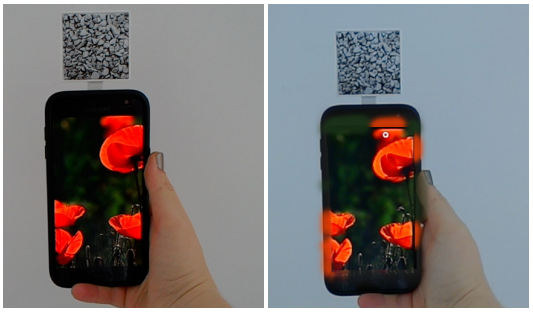
Back of device mirror This prototype displays an AR copy of the handheld device's screen on the back of the device. It allows a user to keep track of what is displayed on their device's screen. The typical scenario involves showing a digital scene to a friend. When the user faces the back of his handheld device, they keeps a live view of what is displayed on the screen. For now, the duplicated AR view is not interactable but this is an improvement that could be implemented in the future. In our design space, this application falls within the following cell: *Behind* × *Duplicate*. Figure 2.9 illustrates this application.

Task Selector This application allows a user to switch between the different apps that are opened on their handheld device. A stack of all the opened apps is displayed on top of the smartphone. When a user looks at this stack (*i.e.* the gaze cursor points at the stack), stacked apps get spread, giving a detailed view to the user. The user can freely



a) b) c) d)

Fig. 2.10: Interaction steps with the *Task selector* prototype: a) all the opened apps form a stack; b) the user gazes at the stack; c) the user selects the calendar app; and d) the smartphone's view gets replaced with the calendar app.



a) b)

Fig. 2.11: On a), the user watch a video without using the *Ambilight* prototype, while in b) the *Ambilight* prototype is used to create a halo around the smartphone's screen.

move their gaze cursor in this menu from an app to another. By performing a *tap gesture*, the user selects a screen, which changes the currently running task of the smartphone. Then the apps get re-arranged into a stack. In our design space, this application belongs to cell *Side* × *Add interaction*. Figure 2.10 illustrates it. This prototype refines and actually implements an idea that had remained conceptual in the VESAD [148] article.

Ambilight The *Ambilight* application allows the user to have a smoother virtual transition between the mobile device's screen and the environment. It got inspired by an actual technology for TVs that relies on LEDs that are located on the side of the TV to display a colored halo around it. Here, we provide a fully virtual Ambilight by displaying a color gradient in AR around the smartphone. To achieve that effect, we capture the handheld device's screen as an image and apply a gaussian blur to it. Then we remove the center of the image to let the smartphone's screen appears. In our design space, it falls within the *Side* × *Enlarge* cell. This application does not involve any interaction from the user, it is purely visual. Figure 2.11 illustrates it.

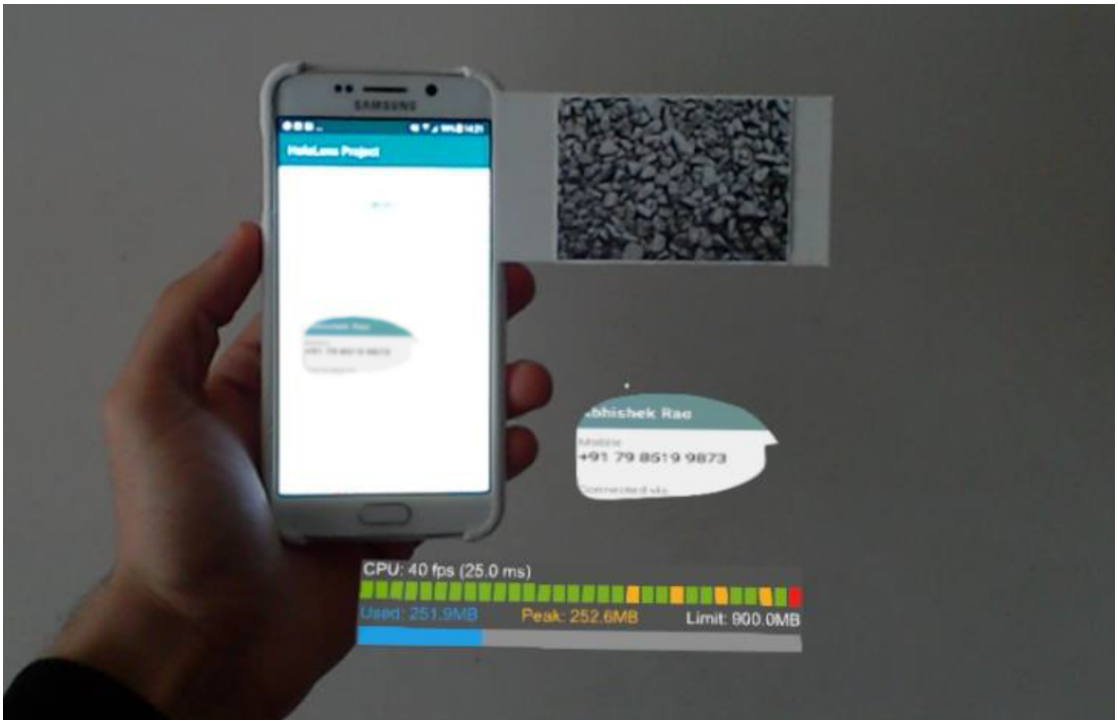


Fig. 2.12: HoloLens 1 view of the *AR-extended screenshots* application. To generate such AR content, the participant has to: 1) click on a "crop" button, on the smartphone's screen; 2) use the touch interface to circle the area of interest in the picture; 3) click on a "done" button to send this area to the HoloLens.

Internship projects

AR-extended clipboard I had the opportunity of helping to supervise the 24H internship of Mariam Diallo and Mamoun Elkhettar, both students from Polytechnique (France). They designed and implement a prototype of an application for saving selections of a smartphone's screen in AR, using a HoloLens 1. To do so, they developed two applications. They used Android Studio for the smartphone application and Unity for the HoloLens application. Both could communicate using a TCP/IP protocol. The smartphone was equipped with a marker, detected by the HoloLens using Vuforia. The *AR-extended screenshots* make it possible for a user to offload content displayed on the smartphone's screen in AR. When entering the smartphone's application, the user can browse their phone's gallery. Once the picture is selected, the user can select the area of interest of this picture, by circling it using the touch interface. Once the finger leaves the display, the picture is shown as cropped and can be offloaded by clicking on a button. This project falls within the *Enlarge* $\times$ *Side* cell of our design space.

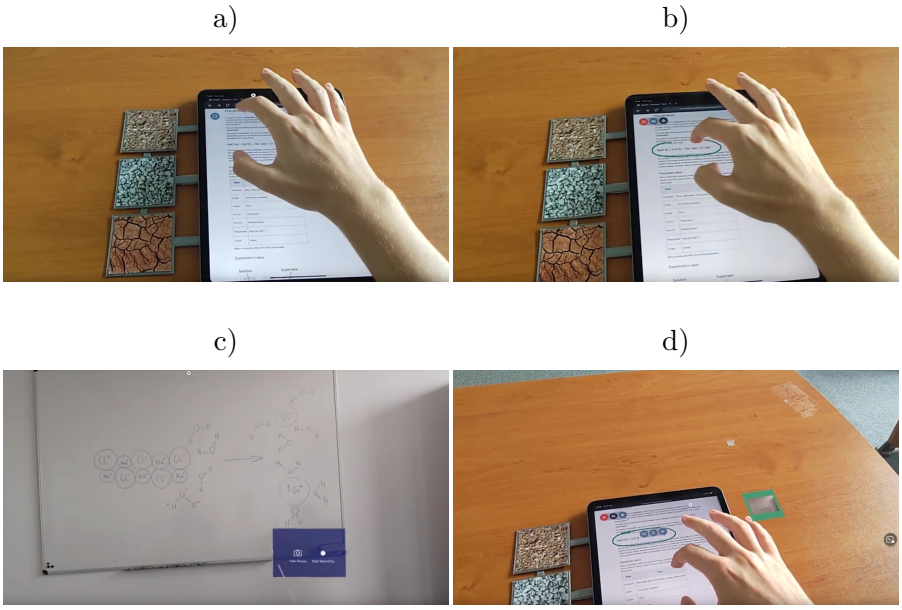


Fig. 2.13: Example scenario of the *AR-extended annotations* application. Here, the user: a) selects an area to annotate on a text file; b) decides to annotate the selected area with a picture; c) takes a picture from the HoloLens' point of view by clicking on the corresponding AR button; d) returns to the tablet, observing the new annotation displayed on the right of the device's screen.

AR-extended Annotations I also co-supervised the internship of Francesco Riccardo Di Gioia, Master student from Université Paris-Saclay (France). We have investigated ways to streamline content annotation on a handheld device by combining it with an Augmented Reality Head-Mounted Display (ARHMD). Francesco designed a full prototype of an application that provides users with annotations offloading in the AR space around a mobile device and also makes it possible to annotate documents not only with traditional means such as text entry and inking, but with content captured on the spot using the ARHMD's camera as well. This prototype works with a Samsung Galaxy A50 or an Ipad Pro combined with a Microsoft HoloLens 2. Regarding software, he developed this prototype using Unity and Vuforia. Both the smartphone and the tablet were equipped with three Vuforia markers, all with the same orientation as their screen to make the Vuforia tracking more robust. This application allows users to select content on different media (*i.e.* 1) Text documents, 2) Maps, and 3) Videos). The selection can be a media unit (*resp.* highlighted text, pins, and frames) or a free-form drawing mark. The prototype allows users to annotate with: 1) typed or handwritten texts, 2) photos, 3) phone screenshots, and 4) videos. Annotations are offloaded in the air, around the device. They are visually linked to their scope as the same color is used for the content selection and the outline of the annotation. These colors are automatically generated by the application, which relies on a color palette for categorical encoding from Color Brewer [144] to make them easily distinguishable. A user study to evaluate this system's usability is in progress.

2.5 AR-enhanced Widgets: Design Considerations

Our design space shows that there are many possibilities for enhancing phone-centric interactions with AR. However, as mentioned earlier in this chapter, the quantifiable pros and cons of distributing a UI across a mobile device and the air are unknown. As a first step toward providing quantitative assessments, we considered the case of standard phone widgets and studied in depth how they could take advantage of AR enhancements.

We first identified three low-level, generic interactions[3] typically performed using widgets that could be at least partially offloaded to the air, either their input, their output, or both: command trigger/item selection; parameter value adjustment; and precise 2D selection. We then prototyped different AR-enhanced techniques, discarding those that we perceived as clearly ineffective. Among the retained techniques, some raised specific questions or required some parameters to be fine-tuned empirically, in which case we ran preliminary experiments to inform these choices. This section reports on the main considerations involved in making design and implementation decisions.

2.5.1 Phone-centric Interaction Space

One of the first decisions we had to make was to bound the input space around the phone. In HMD-centric distributed UIs [218], the phone acts as a controller for AR content that can be anywhere in the user's physical environment. It can be tethered to the phone, or free-floating. In phone-centric distributed UIs, the phone holds the primary content. We thus focus on the direct surroundings of the phone, considering widgets and input areas tethered to the phone.

We adopted the rule of thumb that interaction should take place *not further than 10cm away from the phone*. We found that this distance allowed for comfortable posture and movements, especially when the user's upper arms are along their trunk. It avoids large arm movements in the air, which can be both tiring [76] and socially awkward [2]. It also avoids large eye movements and limits problems of divided attention between the phone and the AR display.

2.5.2 Legacy Bias

Past experience with interactive systems influences the way users apprehend a novel system, making them expect the novel system to work the same way as the systems they are familiar with. Brudy *et al.* [20] warn interaction designers about this so-called *legacy bias* [139] in the context of cross-device interaction. This led us to consider how people interact with smartphones, and find ways to *augment* their experience rather than

[3] Navigation interactions (scrolling, panning, zooming) represent another category, that usually does not involve widgets on a smartphone and has been investigated elsewhere (see, *e.g.*, [85]).

radically change it. The three generic interactions we identified have a straightforward correspondence with UI widgets on smartphones:

- command trigger and item selection are typically achieved with buttons, checkboxes, or menus;
- parameter value adjustments are typically achieved with sliders;
- precise 2D selection is achieved by pointing at a precise location inside widgets such as images, text areas, maps.

Keeping the issue of legacy bias in mind, we favor mid-air interactions which are as close as possible to their touch-only counterparts, striving for a unified set of interaction techniques across modalities. In our design, offloading widgets from the phone to the air does not noticeably change their appearance. Interaction with these widgets is directly inspired by the paradigm developed for the Microsoft Hololens 2 for hand input. Users *touch* discrete widgets with their index tip, and perform continuous input by adopting a *pinch* posture and moving their hand (air-tap-and-hold). These can be seen as the mid-air counterparts to *tap* and *slide* interactions on the phone's screen.

2.5.3 Three-state Mid-air Input

Touch input is direct, with focus implicitly given to the widget being touched. Mid-air input is more indirect. It calls for a three-state model that will enable the selection of a widget before it gets manipulated. Figure 2.14 illustrates the different hand postures that we use to distinguish the three states from Buxton's model of graphical input [26]: `Out of Range`, `Tracking` and `Dragging`. This is a refined version of the air-tap-and-hold interaction implemented on Microsoft HoloLens devices, where the system only supports two states based on hand posture: *pinch* for `Dragging` and *release* for `Out of Range`. We introduce an intermediate posture, *hover*, to enable the third state (`Tracking`) triggered when the thumb and index fingers are close to – but not touching – one another (Figure 2.14-b).

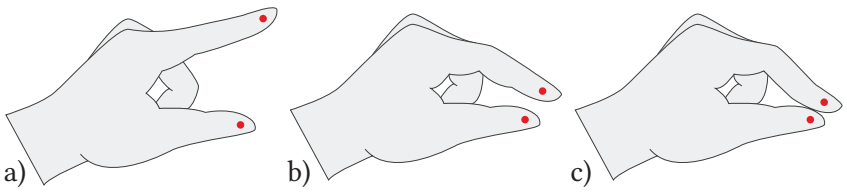


Fig. 2.14: The interpretation of users' hand movements in the air depends on the distance Δ_{TI} between the thumb's tip and the index finger's tip. a) `Out of Range` ($\Delta_{TI} > 4$cm): hand movements have no effect; b) `Tracking` (2cm $< \Delta_{TI} <$ 4cm): hand movements control which widget is in focus; c) `Dragging` ($\Delta_{TI} < 2$cm): hand movements actually control the widget in focus.

When adopting a *pinch* or *hover* hand posture, users control an invisible pointer located halfway in-between the thumb's and index finger's tips. The *hover* posture is used to select which widget has focus. When the pointer enters a widget, that widget gets highlighted with a blue halo. If the user brings his fingers closer to actually adopt a *pinch* posture, the widget will then own focus and subsequent hand movements will control it. A widget that already has focus can be controlled by initiating an air-tap-and-hold interaction either on the widget itself or anywhere in empty space.

This third state also enables us to provide users with a straightforward way to customize how the UI is distributed around and on the phone. An air-click event[4] immediately followed by an air-tap-and-hold interaction[5] actually grabs the highlighted widget. Users are then free to drop it wherever they want by releasing their *pinch* posture. This works both for widgets displayed on the phone and widgets that are already in the air.

2.5.4 Mid-air Input Accuracy

The three-state model above addresses the issue of expressive power, but it does not address that of input accuracy. While indirect mid-air input can contribute to improving accuracy (less occlusion, larger widgets), the absence of physical support and haptic feedback mean that movements are typically less precise (because of, *e.g.*, hand tremor). Furthermore, from a design perspective, movements in the air need to be segmented in order to isolate control gestures from other movements. With the three-state model, we rely on the *pinch* posture for segmenting continuous input from other movements. But releasing control necessarily entails relaxing this *pinch* posture. As opposed to lifting up a finger from a touchscreen, relaxing a posture is a movement, which thus cannot be detected immediately. The system needs to collect a sufficient portion of the relaxing movement before actually releasing control, which inevitably causes an involuntary movement of the cursor. However, since cursor control is indirect with mid-air input, we can account for such inaccuracy by adjusting the CD gain. Setting it to a large-enough value, involuntary movements will not cause the cursor to move.

We ran a preliminary experiment involving six participants to quantify the amplitude of such involuntary movements, which we identify as $\Delta_{release}$. Participants controlled a cursor on the phone with mid-air movements. They had to put it precisely on a very small target (1.5mm), and then switch from a *pinch* to a *release* posture. We measured $\Delta_{release}$, the offset between the target position and the cursor's actual position at the time the system detects this change of hand posture. The design, procedure and results are detailed in supplemental material [6]. Averaging all collected values, we obtain $\Delta_{release} = 1.4 \pm 2.2 < 4mm$.

[4] An air-tap event followed by an air-release event less than 500ms later.

[5] The air-tap event should happen less than 500ms after the last air-click event.

[6] Study data are available as supplemental material online at `https://ilda.saclay.inria.fr/ARwidgets4phones/`

We use this 4-mm maximum bound for $\Delta_{release}$ in the design of interactions that involve mid-air input by enabling users to adjust the CD gain up to $4mm : cursorUnit$ so that the inaccuracy inherent to the segmentation of air movements will be compensated for, at least when the CD gain is maximum.

2.5.5 Effective use of the Air as a Display

The above experiment was coupled with another one aimed at gathering empirical data about users' widget placement preferences. Indeed, widgets can be placed anywhere in the air around the phone. But some sides will generate occlusion, or require adopting uncomfortable postures, depending on which hand is holding the phone. In this second preliminary experiment, participants had to interact with buttons and sliders displayed in the air on each of the phone's four sides. The design, procedure and results are detailed in supplemental material.

Overall, a majority of participants (5) ranked the right side as their preferred location for interacting with *Sliders*, the top side coming second. Results were more nuanced for *Buttons*. Half of the participants ranked the top side first, and the other half the right side. The area below the phone requires users to adopt uncomfortable postures, at least with the limited vertical field of view of current ARHMDs. Our findings are consistent with the recommendations from Normand and McGuffin [148], who report that some study participants had trouble interacting with the AR content around the phone when they had to cross their arms (for instance when they had to interact with virtual elements on the phone's left side using their right hand).

2.6 AR-enhanced Widgets: Design and Comparative Evaluation

We designed AR-enhanced widgets that cover a representative sample of phone interactions: *Command trigger* (E_1), *Parameter adjustment* (E_2), and *2D selection* (E_3). For each of these interactions, we designed several variations of the corresponding widget: one that offloads the widget to the air, one that extends the input space to the air, one that does both. Specific design choices were guided by the considerations detailed in the above section. We systematically compared those designs against state-of-the-art touch-only input techniques in an exploratory study.

We conducted one study per interaction. The three studies were conducted sequentially, always in this order. Participants had to take a break of at least 15 minutes between them. During these breaks, they were asked to rate each technique along all six NASA-TLX dimensions, and then to rank the techniques according to their preferences.

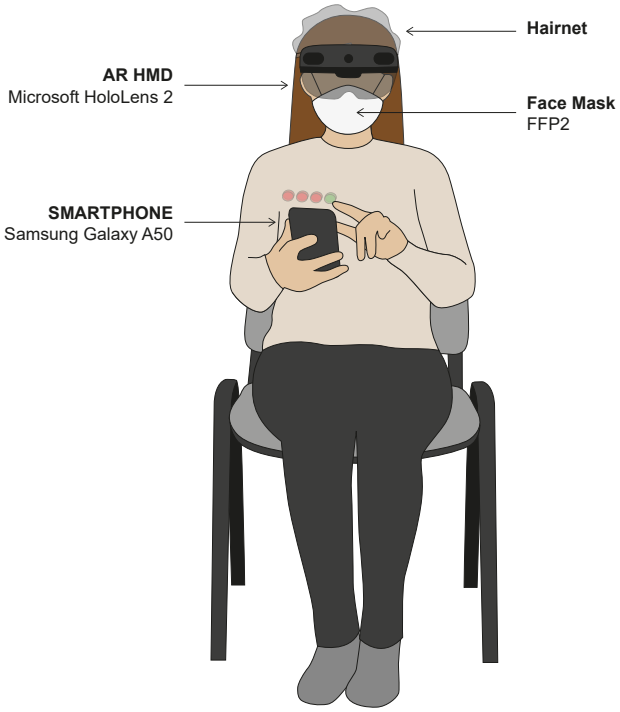


Fig. 2.15: Illustration of a participant during the experiment.

Participants and Apparatus 12 participants (5 women, 7 men, 32 ± 11.1 year-old on average), all right-handed, took part in the study.[7] They were seated on a chair without armrest. We used a Samsung Galaxy A50 and a Microsoft HoloLens 2. The two devices communicate on a dedicated wireless network using TCP. We track the phone's location with Vuforia,[8] using three markers offset by 8cm from the phone's top edge, as illustrated in Figure 2.17.

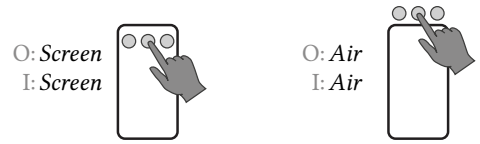
2.6.1 *Experiment E_1 Command Trigger*

In this first experiment, participants are asked to push buttons, either on the phone or in the air around the phone. We describe techniques based on where the output (O) and input (I) take place, respectively. Technique *Screen - Screen*, our baseline, displays buttons on the phone and expects touch input. Technique *Air - Air* displays buttons in the air and expects direct input in the same place.[9]

[7] **About COVID-19 and experiments**: all experiments reported in this article were run during the summer of 2020. Throughout all studies, we recruited participants from a very restricted pool of 12 volunteers: six relatives of the first author who live in the same house, and three volunteer co-workers. The sanitary protocol involves hand disinfection, device cleaning, wearing hair protection under the ARHMD and a FFP2 safety mask. Figure 2.15 illustrates the setup for the experiment.

[8] `https://developer.vuforia.com`

[9] A button is a graphical component that implements the `HandInteractionTouch` script from the Microsoft Mixed Reality Toolkit (MRTK). To trigger an event, the index finger's tip must enter and then leave the component.



Offloading buttons to the air has the benefit of freeing screen real-estate on the phone. But the lack of haptic feedback with air buttons will likely make input more difficult. Furthermore, many users have a lot of prior experience with traditional touch buttons. However, we do not know how much this impacts user performance with air buttons.

Task Three circular buttons, horizontally aligned, with a spacing of 3mm between them, are displayed either 5mm below the screen's top edge (*Screen - Screen*), or 5mm above it (*Air - Air*). Button diameter varies according to task DIFFICULTY: 2cm (*Low*), 1.5cm (*Medium*) or 1cm (*High*). When a button gets highlighted, participants have to select it as fast and as accurately as possible. Buttons in a sequence are always highlighted in the same order: left, right, middle.

Design We follow a within-subject design with two factors: TECHNIQUE $\in$ {*Screen - Screen*, *Air - Air*} and DIFFICULTY $\in$ {*Low*, *Medium*, *High*}. Trials are blocked by TECHNIQUE. Block presentation order is counterbalanced across participants with a Latin Square. Each TECHNIQUE block consists of 12 trials, *i.e.*, four replications per TECHNIQUE $\times$ DIFFICULTY. The presentation order of those 12 trials is counterbalanced across blocks and participants with a Latin Square. Each block starts with three practice trials presented in order of increasing difficulty.

Results Our main measure is the *Time* between two button activations. We filter out interactions with the first (leftmost) button in a sequence as there is no control over the initial position of the participant's hand. We observe a large difference between techniques, with average time 0.4$\pm$0.25s for *Screen - Screen* and $\in$ 1.2$\pm$0.8s for *Air - Air*.

Before conducting any analysis, we check that the collected data follow a normal distribution with a Shapiro-Wilk test. As we observe a violation of normality in one of the conditions (*Screen - Screen* $\times$ *Medium*, $p<0.05$), we log-transform our measure data to avoid having a strong violation of the normality test.[10] An analysis of variance (anova) of TECHNIQUE $\times$ DIFFICULTY on *Time* reveals a simple effect of both TECHNIQUE ($F_{1,11}=181$, $p=0.001$, $\eta^2_G=0.9$) and DIFFICULTY ($F_{2,22}=12$, $p=0.001$, $\eta^2_G=0.13$). *Screen - Screen* is significantly faster than *Air - Air* (0.4s vs 1.28s on average), and the three levels of difficulty are all different from each other ($p's<0.05$).

[10] A quick sanity check shows that significant effects and differences are overall the same when running the anova and post-hoc tests on the non-corrected *Time* measure. The log correction tends to make differences more nuanced but, overall, the general observations remain the same.

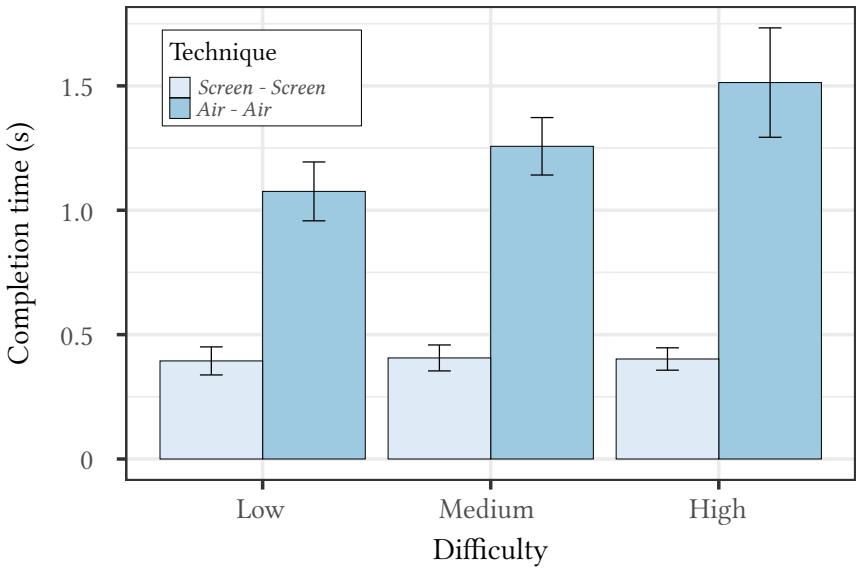


Fig. 2.16: Command Trigger experiment (E_1): completion time by DIFFICULTY × TECHNIQUE. Error bars represent 95% confidence intervals.

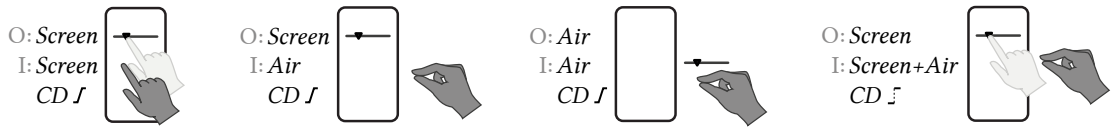
Comparing the techniques' relative performance across participants also reveals a high variability with, *e.g.*, one participant performing 4.82 times better with *Screen - Screen* than with *Air - Air*, and another one 1.88 times only. Interestingly, the participant whose relative performance difference between techniques is the smallest has strong experience with AR headsets. This suggests that users might be able to trigger buttons in the air relatively quickly, but this seems to require some training, and even then performance with on-screen buttons remains significantly better.

Qualitative assessments based on NASA-TLX scales are in line with quantitative observations: average grades on all scales are in favor of *Screen - Screen*. Wilcoxon signed-rank tests reveal that all these differences between techniques are significant (p'$s < 0.05$ and $Z's \in [-2.6, -2.4]$), with the exception of the *frustration* scale ($p = 0.6$). Two participants found interacting in the air *fun* (2 participants), but participants mostly mentioned the advantages of interacting on screen, such as *high reliability* (6 participants) and *haptic feedback* (5). They also reported that touch requires *low amplitude movements* (8) compared to air input, which requires traversing the button and then exiting it with movements that are large-enough to be tracked reliably. Finally, only two participants preferred *Air - Air* over *Screen - Screen*.

Overall, *Screen - Screen* outperforms *Air - Air* in terms of both completion time and comfort. These results are not in favor of offloading phone buttons for activating discrete commands.

2.6.2 Experiment E_2 Parameter Adjustment

In the second experiment, participants had to select a numerical value using a slider. We include high precision tasks which involve value ranges that exceed typical smartphone screen resolution – causing quantization problems [5] – and users' motor abilities in terms of fine-grained control. The touch-only baseline, *Screen - Screen* (CD ⨍), implements OrthoZoom control [6]. This enables participants to perform very precise adjustments, with sub-pixel accuracy. We compare this baseline to three AR-enhanced slider designs (see also Figure 2.17) that vary in where output (O) and input (I) take place, and in how users control input precision (CD):



Screen - Screen **(**CD ⨍**)**. Users interact with OrthoZoom sliders as they do with traditional touch sliders. The only difference is that input precision increases progressively (⨍) as the orthogonal distance between the slider and the finger increases. CD gain is set to 1:1 when the finger is less than 2cm away from the slider's track, and linearly increases up to 1:$sliderUnit_{mm}$ at 10cm from the slider, with $sliderUnit_{mm}$ being the size of a slider unit in mm. Above this maximal distance, a 1mm-displacement thus corresponds to a 1-unit adjustment on the slider.

Screen - Air **(**CD ⨍**)**. The slider is still displayed on the phone, but input takes place in the air. The slider is manipulated with air-tap-and-hold interactions (Section 2.5.3). As with OrthoZoom, input precision is a function of the distance between the fingers' location when the interaction started and the current fingers' location. But in this case, the distance considered is different: precision increases progressively (⨍) as the hand moves away from the phone's plane towards the user's head. CD gain is set to 1:1 when that distance is short (<2cm), and linearly increases up to 4: $sliderUnit_{mm}$ when it reaches 10cm. Above this 10-cm distance, the hand movement must be of at least 4mm to change the slider's value. As discussed in Section 2.5.4, this gives the tolerance required to enable users to release mid-air pinch postures safely without changing the slider's value. The idea of gaining precision for sliders and scrollbars as the distance in the air increases was mentioned in previous work about tabletop interaction [34, 119]. But to our knowledge, it has not actually been tested. Harrison and Hudson [70] explore a similar idea for radial selection on a smartwatch, but do not report how CD gain is adjusted, and study their technique in a very different situation, where participants used an external hardware button attached to a table to validate selections.

Air - Air **(**CD ⨍**)**. The slider is displayed in the air, on the right-hand side of the phone. Interaction is exactly the same as with *Screen - Air* (CD ⨍).

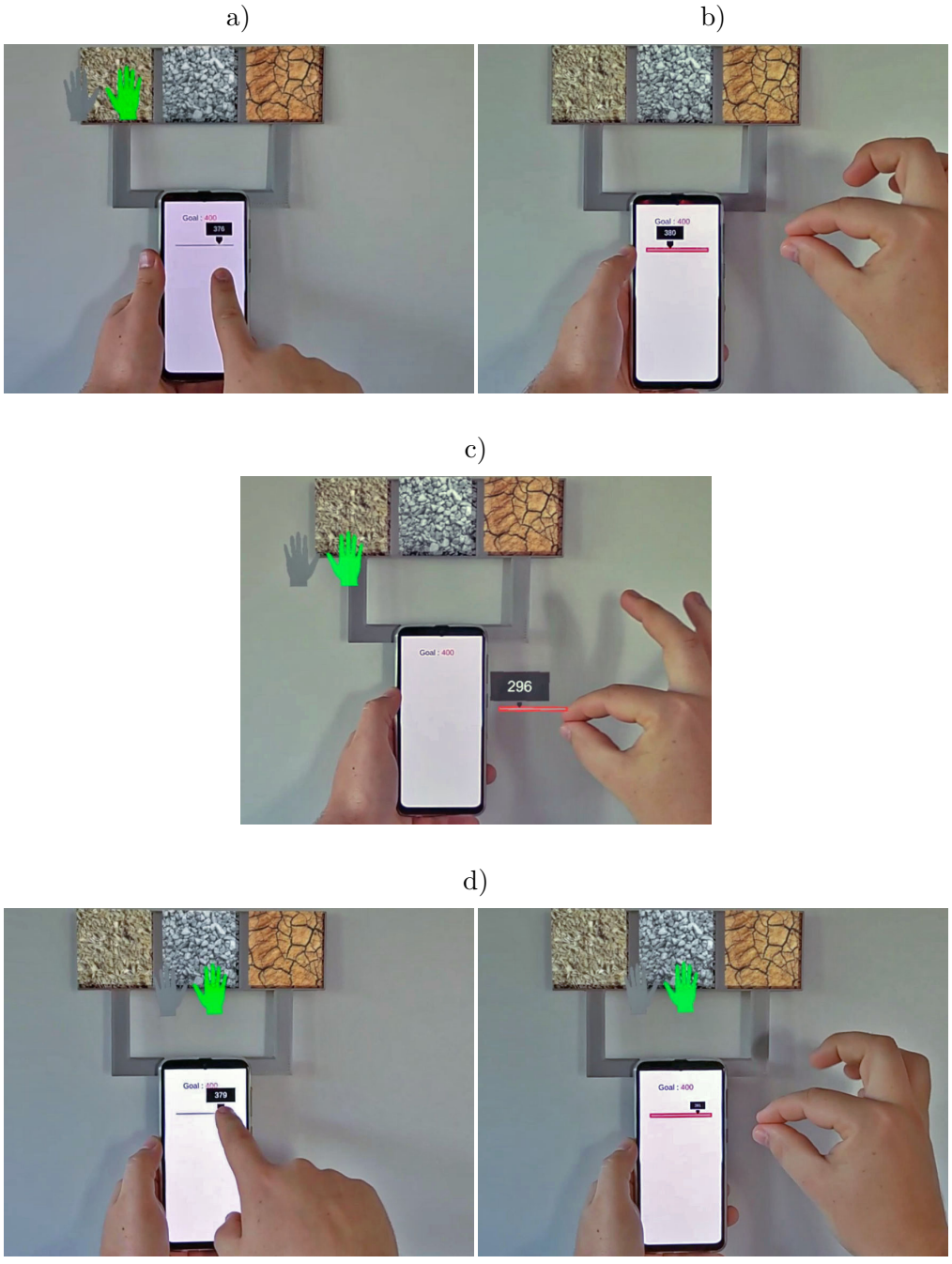


Fig. 2.17: The four techniques tested in Experiment E_2: a) *Screen - Screen* (CD ⌐), b) *Screen - Air* (CD ⌐), c) *Air - Air* (CD ⌐), d) *Screen - Screen+Air* (CD ⌐), performing a coarse selection on the touchscreen (left), then a precise adjustment in the air (right). The two hand icons provide basic feedback about HoloLens left & right hand tracking status and never interfere with widgets displayed in the air.

Screen - Screen+Air (CD ⌐⌐). The slider is displayed on the phone. Input is hybrid, taking place both on the screen and in the air. Users perform coarse adjustments with slide interactions. They can also perform fine adjustments with air-tap-and-hold interactions. There are only two levels of precision (symbol ⌐⌐ means *dual-precision*). CD gain is set to 1:1 for finger slides on the touchscreen, and to 4: $sliderUnit_{mm}$ for movements in the air.

Design properties Each of these four designs has its own advantages relative to the others, but it is difficult to infer their overall performance ranking. First, screen input benefits from haptic feedback, and from the strong experience users have acquired with touch devices. This might be an advantage in favor of *Screen - Screen* (CD ⌐). Second, some designs are consistent regarding where input and output take place, while others are not. Consistency might be preferable and thus be an advantage for *Screen - Screen* (CD ⌐) and *Air - Air* (CD ⌐). But, at the same time, their control requires continuously adjusting two degrees of freedom (slider value, CD gain). They might thus be more cognitively demanding than a clear distinction between only two levels of precision (coarse and fine). In that respect, *Screen - Air* (CD ⌐) might have some advantages over the other three techniques. In the end, these pros and cons make it difficult to know the cost of offloading input, output, or both in the air for slider control.

Task A 5cm horizontal slider is displayed on the phone for all *Screen - ∗* designs (11cm above the bottom edge of the screen) and in the air for the *Air - Air* design (offset 2cm from the phone's right edge). Depending on task DIFFICULTY, the slider has 100 (*Low*), 500 (*Medium*) or 1000 (*High*) units. The value to select, GOAL, is displayed in the upper part of the phone. Participants have to select the GOAL as fast as possible. The trial ends as soon as participants select the goal value (no finger on screen, hand in *release* posture). The slider knob's size depends on the current CD gain (half its default size at maximum precision), providing feedback to users about input precision.

Design We follow a within-subject design with two primary factors: TECHNIQUE ∈ {*Screen - Screen* (CD ⌐), *Screen - Air* (CD ⌐), *Air - Air* (CD ⌐), *Screen - Screen+Air* (CD ⌐⌐)} and DIFFICULTY ∈ {*Low*, *Medium*, *High*}. Trials are blocked by TECHNIQUE. Block presentation order is counterbalanced across participants with a Latin Square. We introduce a GOAL factor that can take four possible values for ecological purposes. It corresponds to the value participants are instructed to select. Each TECHNIQUE block consists of 12 trials, *i.e.*, one per TECHNIQUE × DIFFICULTY× GOAL. Presentation order of trials within a block is counterbalanced across blocks and participants with a Latin Square. Each block starts with three practice trials, presented in order of increasing difficulty.

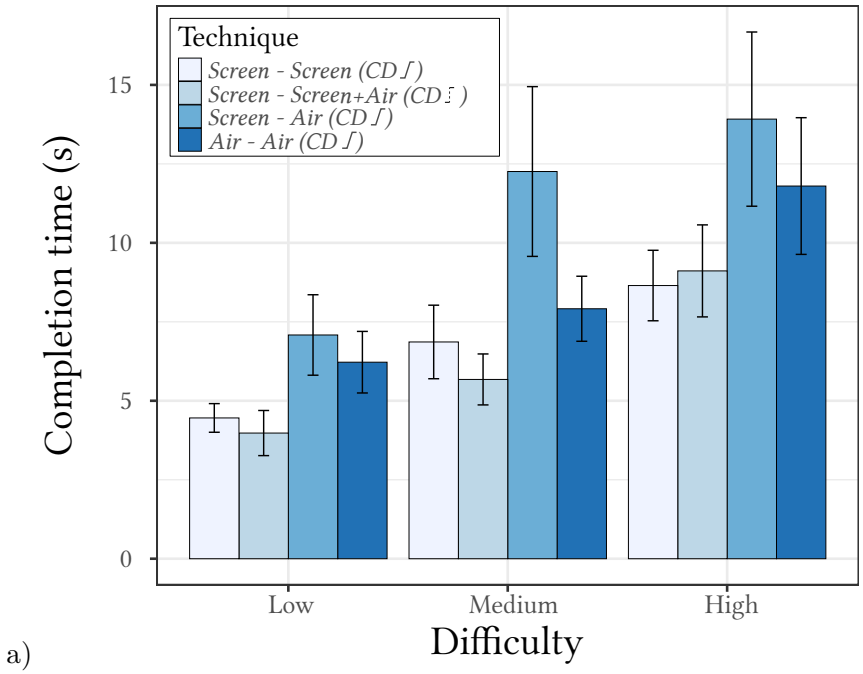
Results We first check the normality of the collected task completion *Time* measures with Shapiro-Wilk tests. We observe that the data are not normally distributed under a few conditions (*Screen - Screen+Air* (CD ⌐⌐) × *Low*: $p<0.001$, *Screen - Screen+Air*

(CD ⌐) × *Medium*: $p<0.05$ and *Screen - Screen* (CD ⌐) × *Medium*: $p<0.05$). Applying a log-transformation solves the issue. We thus perform our analyses on a log-transformed time measure. An anova of TECHNIQUE and DIFFICULTY on *Time* reveals a simple effect of both TECHNIQUE ($F_{3,33}=23$, $p=0.001$, $\eta^2_G=0.5$) and DIFFICULTY ($F_{2,22}=118$, $p=0.001$, $\eta^2_G=0.6$). Unsurprisingly, all levels of DIFFICULTY significantly differ from each other: completion time increases as the task gets more difficult (Figure 2.18-a). Regarding differences between techniques, pairwise t-tests reveal that all techniques significantly differ from each other with all p'$s<0.01$, except for *Screen - Screen+Air* (CD ⌐) and *Screen - Screen* (CD ⌐), which have a p-value close to 0.05 ($p=0.04$).

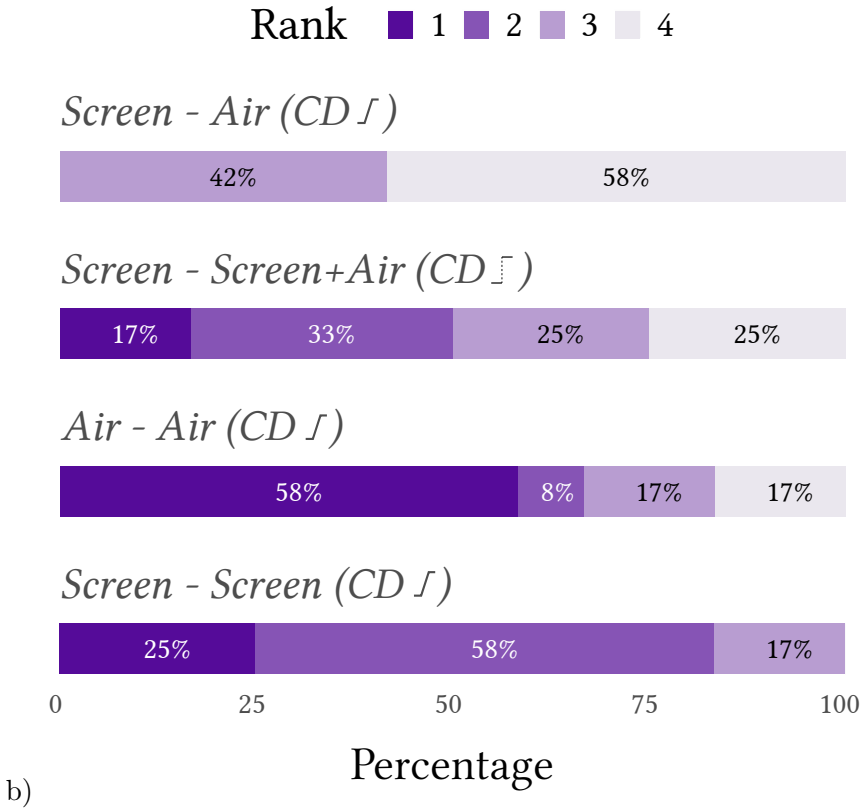
Sorting techniques by performance, we get: *Screen - Screen+Air* (CD ⌐) (6.25s), *Screen - Screen* (CD ⌐) (6.65s), *Air - Air* (CD ⌐) (8.64s), *Screen - Air* (CD ⌐) (11.1s). Compared to experiment E_1, the drop in performance when offloading input and output in the air is much lower with sliders than with buttons. Interestingly, *Screen - Screen+Air* (CD ⌐) actually performs slightly better than *Screen - Screen* (CD ⌐), the OrthoZoom touch-only baseline. This suggests that air control can even be better than touch for indirect control. However, this is only the case when air control is implemented according to a dual-precision strategy. On the opposite, continuous precision level control is less efficient in the air than on screen.

Qualitative assessment. We perform Wilcoxon signed-rank tests to compare pairs of techniques along each of the NASA-TLX scales. The difference between *Screen - Air* (CD ⌐) and *Air - Air* (CD ⌐) is systematically significant, except for *physical demand*. Participants' comments reflect this preference for *Air - Air* (CD ⌐) over *Screen - Air* (CD ⌐): six of them spontaneously said that they *like* interacting in the air; and five of them reported having issues of *divided attention* when output is on the phone and input in the air, *i.e.*, when there is a mismatch between where input and where output take place. *Screen - Air* (CD ⌐) and *Screen - Screen* (CD ⌐) also differ significantly, with ratings systematically in favor of *Screen - Screen* (CD ⌐) on all scales except *mental demand*. Prior experience actually using an equivalent of *Screen - Screen* (CD ⌐) was reported by three participants and might explain this.

Screen - Air (CD ⌐) and *Screen - Screen+Air* (CD ⌐) also differ significantly along all scales but *physical demand* and *effort*. Users found it easier to switch between two levels of precision with *Screen - Screen+Air* (CD ⌐), as opposed to the continuous precision adjustments of the other (CD ⌐). The former strategy (CD ⌐) also avoids some usability problems. One issue raised by four participants about continuous CD gain adjustment (CD ⌐) relates to clutching. When releasing a pinch posture in mid-air to immediately initiate a new one (*e.g.*, to avoid leaving the tracking volume or to adopt a more comfortable position), CD gain is reset to its default value. These participants commented that they would have preferred the system to set the CD gain to its last value before release instead. Implementing such behavior is not trivial however, as it requires distinguishing between clutch actions and actual successive-but-distinct uses of the same



a)



b)

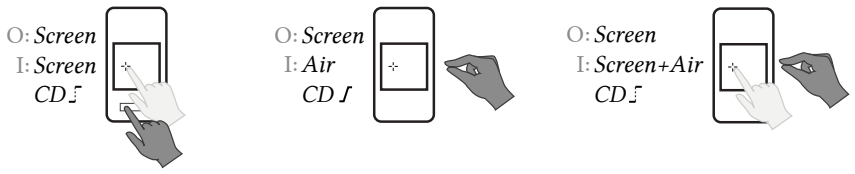
Fig. 2.18: Parameter Adjustment experiment E_2. a) Completion time by DIFFICULTY × TECHNIQUE. Error bars represent 95% confidence intervals. b) Participants' preferences. Each bar represents the rank distribution for one technique.

widget (for which the current behavior makes more sense). It also raises usability issues, such as the higher chance for the phone to *"be in the way"* of the user's hand if resuming slider manipulation too close to the screen. Interestingly, clutching issues are not observed only for air control. Three participants reported sometimes reaching the phone's left or right edge when seeking a high level of precision with OrthoZoom, which forced them to clutch (*i.e.*, releasing control and resuming on the slider's track itself).

Finally, seven participants reported that the switching cost between touch and air is high. Overall, participants' ranking of techniques is not completely in line with actual performance. As Figure 2.18-b shows, preferences are split and the fastest techniques are not systematically the preferred ones. In particular, *Screen - Screen+Air* (CD ⌐) performed best in terms of completion time, but only 17% of participants ranked it first. The novelty effect probably played in favor of *Air - Air* (CD ⌐), which got ranked first by more than half of the participants. On the contrary, there is a clear consensus against *Screen - Air* (CD ⌐).

2.6.3 Experiment E_3 2D Selection

In the third experiment, participants have to point at a precise location in a 2D workspace. Output (O) necessarily takes place on the phone, as the display resolution of current ARHMDs is too low to render 2D targets small-enough for our purposes. We thus focus on where input (I) takes place. Mid-air input has the potential to offer an occlusion-free alternative to touch for precise pointing tasks such as, *e.g.*, positioning a caret in a text or delineating a region in an image or a map. But it will also likely suffer from significant precision issues. We compare input performed in the air to a touch-only technique and to a hybrid technique:



Screen - Screen (CD ⌐) serves as our baseline condition. Direct touch in the 2D canvas lets users position the pointer coarsely. They can then control it indirectly but much more precisely (higher CD gain) by initiating a slide gesture from a smaller rectangular area at the bottom of the screen (2cm above the edge). This is a direct adaptation of a technique available on Android and iOS smartphones for caret positioning, invoked by dwelling on the soft keyboard's space bar or by applying a higher force level anywhere on the keyboard (with pressure-sensitive screens).[11] Our adaptation behaves differently though, changing the CD gain to enable very high precision pointing. The original technique found on

[11] We set the dimensions of this area according to the dimensions of the space bar on the phone used for the experiment (31.77mm×7.75mm), offsetting it 1cm upward to make room for downward movements.

most smartphones keeps a 1:1 CD gain when entering indirect pointing mode.[12] Thus, it does address finger occlusion issues, but it does not enable high-precision pointing. Our implementation is a dual-precision version (CD ⌐) which sets CD gain to 1:$precision_{mm}$ when entering indirect mode, where $precision_{mm}$ represents the size of the smallest element that users might want to acquire. In this experiment, we set it to 0.5mm, meaning that a 1mm finger displacement moves the pointer by 0.5mm.

Screen - Air **(**CD ⌐**)** is similar to *Screen - Air* (CD ⌐) for sliders (in Experiment E_2), but enables 2D selection. Users control the cursor with air-tap-and-hold interactions. Precision gets higher as the hand moves away from the start position of the air-tap-and-hold gesture. The CD gain is set to 1:1 when the distance is low (<2cm), and progressively increases up to 4: $precision_{mm}$ at a distance of 10cm and above. In our study, users are able to point with sub-millimeter precision ($precision_{mm}$ = 0.5mm). Such as the *ARPads* presented in the first chapter of this thesis, this technique implements indirect 2D input with movements in the air. It is conceptually close to the 1-Button-Simultaneous technique from Jones *et al.*'s study [85], which compared different designs for pan & zoom navigation using mid-air gestures around the phone. Although the three input dimensions are not exactly the same in the two studies, their technique and ours are both based on integral and continuous input in a 3D mid-air volume, with movements in the x-y plane for controlling a 2D space (pan *vs.* cursor position) and movements along the z-axis for *scaling* this 2D input (zoom *vs.* CD gain).

Screen - Screen+Air **(**CD ⌐**)** is similar to *Screen - Screen+Air* (CD ⌐) for sliders, but enables 2D selection. Users perform coarse adjustments with direct touch input, and fine-grained adjustments with air-tap-and-hold interactions. Precision-level control is dual (CD ⌐), with the CD gain set to 1:1 for touch input on the phone's screen, and to 4: $precision_{mm}$ for mid-air input.

Design properties Again, each of these designs have supposed advantages, but it is difficult to anticipate their comparative performance. The difference between *Screen - Screen* (CD ⌐) and *Screen - Screen+Air* (CD ⌐) mostly lies in the input modality. *Screen - Screen* (CD ⌐) might benefit from touch input properties such as haptic feedback and prior experience. But at the same time, in the above-mentioned study by Jones *et al.* [85] the 1-Button-Simultaneous – which resembles *Screen - Air* (CD ⌐) – performed as well as multitouch gestures did. This latter result suggests that offloading the control in the air with a continuous control strategy might not have a high cost. The dual-precision control in *Screen - Screen+Air* (CD ⌐) could make this cost even lower considering how it improved air control for parameter adjustment in experiment E_2.

Task Participants have to successively acquire eight circular targets laid out on the phone's screen following the ISO 9241-9 standard [82]. Depending on task DIFFICULTY,

[12] Some smartphones implement a transfer function that makes the CD gain decrease as the finger accelerates to enable coarse adjustments from this mode.

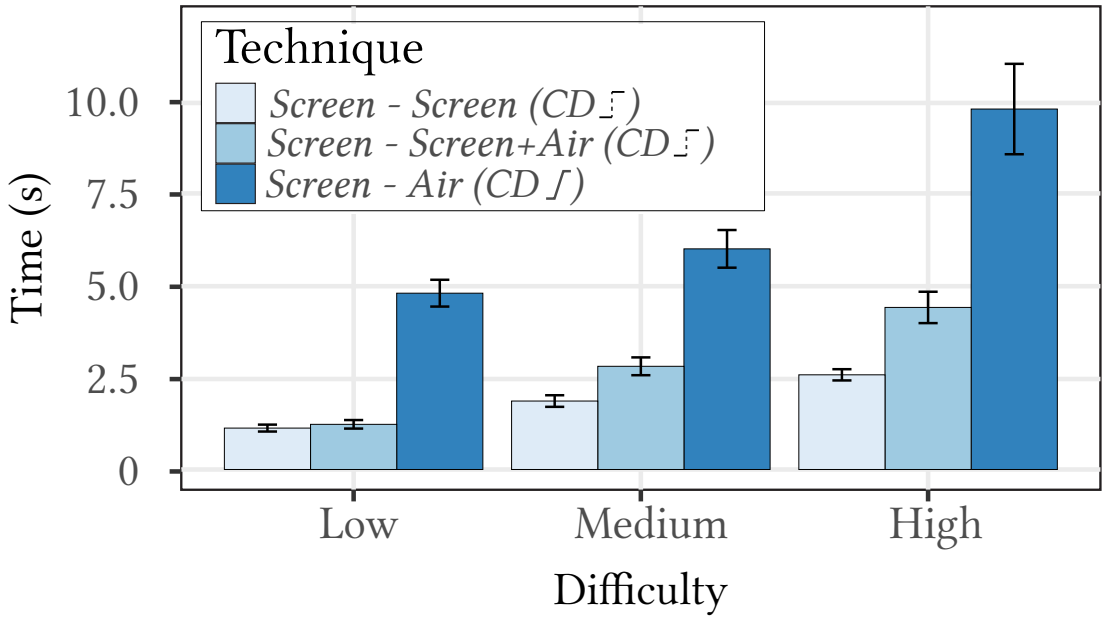


Fig. 2.19: 2D Selection experiment E_3. Completion time by DIFFICULTY × TECHNIQUE. Error bars represent the 95% confidence interval.

target diameter is 4mm (*Low*), 2mm (*Medium*) or 1mm (*High*). The next target to acquire is colored black, while other targets are colored grey. As targets can be very small, the background turns green when the crosshair cursor is inside the target. As soon as participants select the black target (no finger on screen, hand in *release* posture), the next target turns black. Cursor size is inversely proportional to the current CD gain, providing participants with a rough indication of its value, as in Experiment E_2.

Design We follow a within-subject design with two primary factors: TECHNIQUE ∈ {*Screen - Screen* (*CD*), *Screen - Screen+Air* (*CD*) and *Screen - Air* (*CD*)} and DIFFICULTY ∈ {*Low*, *Medium*, *High*}. Trials are blocked by TECHNIQUE. Block presentation order is counterbalanced across participants with a Latin Square. Each trial is a series of eight pointing tasks, and is replicated twice. A TECHNIQUE block consists of 6 trials (*i.e.*, 6×8 pointing tasks). Presentation order for trials within a block is counterbalanced across blocks and participants with a Latin Square. Each block starts with three practice trials, presented in order of increasing difficulty.

Results Before running analyses, we filter out the first pointing task in each series of eight, as the initial cursor position is not controlled when a series starts. We additionally remove one outlier that corresponds to a trial during which the HoloLens actually stopped working for a few seconds. Shapiro-Wilk tests do not reveal any violation of normality. An anova of TECHNIQUE and DIFFICULTY on *Time* reveals a simple effect of both TECHNIQUE ($F_{2,22}=42$, $p=0.001$, $\eta^2_G=0.65$) and DIFFICULTY ($F_{2,22}=39$, $p=0.001$,

$\eta^2_G = 0.41$), as well as an interaction effect ($F_{4,44} = 6$, $p = 0.001$, $\eta^2_G = 0.14$). As illustrated in Figure 2.19, the difference between *Screen - Screen* (CD ⌐) and *Screen - Screen+Air* (CD ⌐) gets larger as difficulty increases, and is not even significant when DIFFICULTY = *Low* ($p = 0.7$). The third design, *Screen - Air* (CD ⌐), performed very poorly in comparison with the other two. Sorting techniques by performance, we get: *Screen - Screen* (CD ⌐) (1.67s), *Screen - Screen+Air* (CD ⌐) (2.53s), *Screen - Air* (CD ⌐) (6.18s). Offloading input to the air has a performance cost. But this cost is limited if air control is implemented with a dual-precision mode, and could be counterbalanced by the advantages that air control has over touch: no occlusion, and no need to reserve space on screen for toggling between precision modes.

Qualitative assessment. Quantitative measures match participants' qualitative evaluation of techniques, in which they systematically rated *Screen - Air* (CD ⌐) as significantly worse than the other two along all NASA-TLX scales, with the exception of *Screen - Screen+Air* (CD ⌐) for *Mental Demand* ($p = 0.07$). This poor performance of *Screen - Air* (CD ⌐) is consistent with the findings from Experiment E_2. It again proved difficult to control position and input precision with a movement in the air while the output was on phone. Interestingly, this is different from what Jones *et al.* [85] observed. Several elements might be at play. First, their technique was used for multi-scale navigation, which affects the whole graphical scene rather than just the cursor's position and size. The stronger visual feedback associated with their task might be more effective for continuous control. Second, zoom control is relative in their case, while precision control is absolute with *Screen - Air* (CD ⌐). As discussed in E_2, implementing relative control when clutching is not trivial, a problem which does not exist in the case of multi-scale navigation. One last difference is about the delimiter for mid-air gestures. Jones *et al.* rely on a physical button on the phone, which might be faster than the *pinch* posture that we use. However, participants in their study also complained about this physical button delimiter which they found uncomfortable to use.

Finally, participants' preferences are distributed across *Screen - Screen* (CD ⌐) and *Screen - Screen+Air* (CD ⌐) with eight participants ranking *Screen - Screen* (CD ⌐) first, the other four preferring *Screen - Screen+Air* (CD ⌐). They consistently ranked *Screen - Air* (CD ⌐) third.

2.7 Summary of Findings

We compared different distributed UIs for three tasks that are representative of touch interaction with smartphones. While we observe an input performance drop with air controls in many cases, this drop is compensated for in certain situations since offloading widgets to the air has interesting properties. Widgets displayed in the air are readily accessible when they would otherwise be several steps away in the hierarchy. Some phone screen real-estate gets freed up to show more content. We derive the following guidelines from these empirical results.

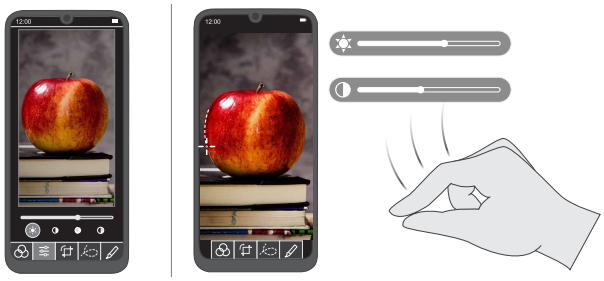


Fig. 2.20: Redesign of the image editing application from Figure 2.1 following the guidelines derived from our empirical results. Buttons remain on the smartphone. Sliders for parameter adjustments performed frequently get offloaded. 2D selection can be performed directly on the phone, or indirectly in the air (dual-precision mode) if occlusion is an issue.

First, *discrete controls (*e.g., *buttons, checkboxes) should not be offloaded to the air.* Users have much trouble activating discrete controls with air gestures, at least with the default MRTK behavior used on the HoloLens 2.

Second, *dual-precision air control (CD ⋮⌐) can effectively complement touch for parameter adjustments and 2D selections.* Our observations reveal that users' performance and preferences are split between touch and dual-precision air control for on-screen widgets. Participants were faster in experiment E_2 with dual-precision air control than with touch control for sliders on screen. In experiment E_3, they performed equally well with touch and air controls (both dual-precision) for low-difficulty cursor positioning (4-mm precision). Air control performance degraded at higher difficulty, but the technique remains interesting nevertheless. First because it supports a three-state model of input which gives interaction designers more expressive power. But also because it is indirect and thus avoids visual occlusion issues – which were not operationalized in our task. Occlusion can be partly addressed in a touch-only context by featuring an area on the phone's screen to trigger indirect control as our baseline condition does in experiment E_3, but this takes some screen real-estate. Thus, as touch and air controls can coexist, there are only benefits in supporting both.

Finally, *frequently-used sliders can be good candidates for offloading to the air.* While participants were slower with air sliders than they were with screen sliders, the performance drop of *Air - Air* (*CD* ⌐) illustrated in 2.18 (about 1-to-2 seconds) can be considered reasonable when put in perspective. Touch sliders are faster, but they consume a lot of screen space and accessing them often involves navigating menus (2.1-(a)). Air sliders are slower, but can be accessed very quickly without navigating the menu hierarchy. In addition, they do not generate any visual occlusion, as opposed to high-precision sliders such as OrthoZoom which extends control space beyond the widget's visual footprint. Interaction designers should thus consider this trade-off when making a choice between touch and air sliders for parameter adjustments.

Figure 2.20 illustrates how these guidelines can be applied to the design of our image editing example from Figure 2.1. The toolbar remains on the phone's screen as offloading buttons in the air would cause too much of a drop in performance. Frequently-used sliders, which were accessible after navigating a menu, are offloaded in the air and thus brought to the application's top layer. The second button in the toolbar ( ), which was used to invoke those sliders, is thus no longer necessary. This frees a bit more space in the phone's toolbar to accommodate additional controls or enlarge other buttons. Finally, 2D selection can be performed directly on the phone, or indirectly in the air (dual-precision mode) if occlusion is an issue.

2.8 Limitations and Future Work

Studies as ours necessarily depend on the characteristics of the apparatus. Even using a state-of-the-art ARHMD, there are limitations, related primarily to the field of view and tracking accuracy. Our study gives a sense of the comparative performance of different widget designs with *current* AR technology. As this technology improves, comparative performance results might evolve. Novel sensing and display capabilities might also enable enhanced widget designs. Beyond initial empirical results, our contribution also includes a series of AR-enhanced widget designs, along with a set of tasks representative of phone interactions to test them. These can serve as a basis for future studies.

The observed performance drop of air control relative to touch should be interpreted keeping in mind that we compared AR-enhanced techniques to OrthoZoom and to a dual-precision touch pointer. These state-of-the-art touch-only techniques are hard to beat, even more so given the experience people have accumulated with touch interfaces over the years. Several participants reported that they found touch input more reliable, but also that they were feeling increasingly efficient with air input as they were proceeding with the experiment. It would be interesting to conduct longer studies in order to quantify learning aspects and identify performance envelopes [43]. Follow-up studies should also consider more elaborate tasks to evaluate the impact of visual occlusion and the overhead caused by having to navigate in the widget hierarchy in touch-only conditions.

Because of the covid-19 pandemic, the studies were run with a restricted pool of participants, including the authors themselves. Although this is unusual, we believe that our results have reasonable validity. First, running our statistical analyses without the authors does not change the outcome. Second, our contribution is an exploration of various AR-enhanced widget designs. There is no notion of *the authors' solution* and thus no bias in favor or against any specific design. Our pool also includes both participants with much ARHMD experience and participants who had never even seen one before. Despite this great variability, we were able to observe significant – and sometimes quite large – effects. However, we insist that our results remain exploratory, and should not be generalized without further investigation.

Finally, there are likely opportunities to improve air controls in terms of interaction design. For example, we used the default MRTK behavior for air buttons, but we thought about other mechanisms to activate them. For instance, by crossing them with finger movements coplanar to the phone, whose edges would provide haptic feedback. While participants performed better with sliders displayed on the phone but controlled in the air than with sliders fully in the air, several of them indicated that they preferred the latter. They did not like the input-output mismatch of the former. Adapting a dual-precision strategy for sliders fully in the air could also improve performance.

2.9 Conclusion

To help the wide adoption of AR HMD in users' everyday life, we should work towards advancing relevant use cases. In this chapter, we have discussed the benefits of AR for enhancing other everyday devices such as smartphones. We have first contributed a design space of smartphone-centric applications enhanced with AR. This design space described two dimensions which are the *FUNCTION* of the holographic content and its *LOCATION*. We populated this design space with examples from the existing HCI literature, demonstrating its descriptive power. We also provided novel ideas of AR-enhanced smartphone-centric applications, proving that this design space can also be used to generate new concepts and that AR could bring benefits to many phone-centric interactions. While some of these ideas remain conceptual, some led to actual implementations, demonstrating the feasibility of such applications.

We have also conducted a preliminary investigation, comparing different distributions of input and output between the holographic screen and the device's screen for common smartphone-based widgets. In particular, our study reported on command triggering, parameter adjustment, and 2D selection. All AR-enhanced widgets were compared to their touch-only counterparts. Our results suggest that, while touch-only widgets performed better in most cases, some of the proposing AR-enhanced distributions did reach similar performances, showing the practicability of such distributions. We do not push for the use of AR to offload all smartphone-based widgets, but rather suggest that frequently used widgets could benefit from being permanently accessible in AR, as this would save some time to access them and some valuable screen-estate.

In this chapter, we have provided practical use-cases for AR to complement daily devices as much as valuable guidelines for future AR-enhanced, smartphone-centric applications regarding the UI distribution across the two environments. These suggest that AR could be used on a daily basis, being integrated into users' digital environment.

3. DE-AUGMENTING AUGMENTED REALITY

This chapter is based on a work I have done in collaboration with Caroline Appert, Anastasia Bezerianos, Olivier Chapuis, and Emmanuel Pietriga. It is currently under review for CHI '22.

3.1 Introduction

Introducing AR in users' daily life is not only a matter of appealing use cases. The technology's form and performance should be good enough to be acceptable. Wearable Augmented Reality (AR) devices are improving on several fronts in that regard. Their field of vision is getting wider. Display resolution is increasing. Enhanced tracking capabilities enable better input techniques. Clumsiness is still an issue, however. Helmets and glasses remain overly conspicuous and uncomfortable for people to consider wearing unless necessary. Yet we can anticipate progress on this front too as AR-enabling eyewear technology evolves. Miniaturization and progress in terms of autonomy should eventually make it possible to embed AR technology in prescription glasses. AR displays are also getting integrated in domain-specific pieces of equipment in which they are considered socially acceptable already [99]: motorcycle helmets, safety glasses, hard hats. Near-eye displays [14, 30] can involve radically different technologies such as metasurfaces [108, 103] and other types of contact lenses [150, 153], though these are still in the early stages of development.

As mentioned earlier in this manuscript, both academia and industry are anticipating such improvements and are contemplating an increasingly diverse range of use cases for AR. Beyond the well-known ones [15] such as gaming [157], education [169] or industry [55], use cases now include domestic use [97], therapy [87], emergency management [178, 179], assisting people with visual impairments [104, 131], a large range of sports & leisure activities, spatial orientation assistance at large [28], as well as advertising [151].

Developments on the technological side and on the usage side go hand in hand, and suggest a future in which AR has become pervasive. Becoming inconspicuous, the technology will get socially acceptable beyond the workplace [98, 174]. Becoming practical in many

contexts, it will allow for the actual realization of many of the envisioned use cases, as well as new, unanticipated ones. Removing these barriers to adoption will likely entice people to wear AR on a very regular basis, if not all the time. Perhaps more insidiously, some future forms of AR displays – such as contact lenses – could actually be impractical to put on & off continually, leading their users to keep wearing them even when not actually needed, which might in turn entice them to leave AR turned on.

This can be seen as a virtuous or a vicious circle depending on the perspective we take on it; but one that leads part of society to always-on AR, regardless. Matsuda [121] as well as May-raz & Lazo [122] both propose striking depictions of this in their futuristic short films. Whether we embrace such a future or not, always-on AR raises multiple questions. Many relate to ethics [174, 175, 188], some others relate to practical concerns, and yet others to both. One of them stems from the visual interference that AR can cause with respect to people's perception of, and interaction with, the real world. The more often AR content is shown, the more often it will somehow be "in the way", with consequences that range from mere user annoyance to serious physical injuries. AR designers simply cannot anticipate all situations, and there is a need for users to be in control of what is in their field of vision.

We argue that to achieve proper user agency, user control over AR content display cannot be limited to a simple on/off toggle. Some AR content might be interfering, while other content might be critical to the task at hand or even just convenient to have, thus calling for solutions to *selectively* remove AR elements from the field of vision.

In this chapter, we introduce the concept of *de-augmenting* augmented reality. After positioning our work in the literature, we rationalize the need to give users control over virtual content display through three scenarios. We then build a taxonomy that captures the different facets of de-augmentation. Does the de-augmentation affect a specific area of the user's field of vision or a specific region in the physical world? Is it triggered automatically based on rules or on-demand using, *e.g.*, deictic gestures? Does it hide, composite or geometrically transform the virtual content to mitigate visual interferences? We demonstrate the concept's technical feasibility, using off-the-shelf technology to build a proof-of-concept prototype. Finally, we discuss the many avenues for future work that it opens in terms of both societal concerns (safety, privacy, ethics) and practical concerns (interactive system design).

3.2 Background

De-augmenting augmented reality relates to the more general problem of displaying virtual content so that it effectively coexists with what users need to perceive from the physical environment. While this problem arises across the entire reality-virtuality continuum [128], some questions are of more particular importance in the context of AR. As AR devices typically display content as a semi-transparent overlay on the user's field

of vision, virtual elements can augment the physical world. But they can also visually interfere with the perception of that physical world. Even if not directly interfering from a perceptual perspective, virtual content can compete with cues from the environment (and conversely), causing problems of user attention, interpretation and general awareness of one's physical surroundings. Such issues will be even more critical in mobile contexts of use where the physical environment is highly dynamic and unpredictable, making the coexistence of the physical and virtual difficult to effectively automate. In this section, we give an overview of concepts and techniques from prior work that have inspired our reflections on de-augmentation and informed our taxonomy: user attention in a mobile context; techniques that alter our perception of reality; as well as *augmented virtuality* techniques, that bring some elements from the physical world back in a VR (Virtual Reality) environment.

3.2.1 Mobile Technologies and User attention

Compared to stationary hardware such as desktop workstations, mobile devices are often used in much more dynamic and unpredictable environments. Paying too little attention to the physical surroundings in such contexts can cause safety issues. The mobile technology literature reports on studies about this type of attention tunnelling [198], which has actually been investigated in the specific case of smartphone-based AR applications such as the *Pokémon Go* game. For instance, Gruenefeld *et al.* [65] designed glasses equipped with low-cost LED displays placed on their sides. Those LEDs emit light in the periphery of smartphone users' field of vision to warn them about potential physical hazards. Kang *et al.* [91] analyse a smartphone's camera feed to detect forward obstacles. They test different modalities, finding that audio and tactile cues are more efficient than visual ones. CrashAlert [77] implements a similar concept, this time using a depth camera to detect potential obstacles based on their distance to the phone. Different kinds of visual alerts on the phone were observed to be successful at making users adapt their behavior and avoid obstacles.

While all of the above projects study contexts in which users focus on their smartphone, Jung *et al.* [88] address similar physical safety situations, but focus on users wearing AR headsets in the street. They implement a vehicle position estimation technique that is then used to enhance ground traffic situation awareness, providing visual indications about vehicles that are too close. Regardless of the specifics of how the AR wearer is notified about potentially hazardous situations, these represent one obvious case in which de-augmentation can be beneficial. As discussed later, such situations are not the only ones that can benefit from it.

3.2.2 Altering Reality

Part of the above safety issues when wearing AR displays are due to the AR content either distracting users or actually hiding safety-critical elements in the physical environment. But such visual occlusion is actually not the only source of concern about the perception

of the physical environment. AR also has the potential to alter our perception of reality, which can be a major issue when the display covers the whole field of vision or at least a significant part of it.

AR technologies can display content that is obviously virtual, such as floating windows [48]. Such elements are unlikely to be mistaken for physical elements, but they can hinder the visibility of people or physical objects that users interact with. For instance, the recent HoloDoc system [110] builds on the seminal Digital Desk prototype [210], providing a desk environment where virtual objects coexist with multiple physical ones. The latter can be large, such as printed documents, or fairly small, such as pens and erasers whose manipulation can be hindered because of the visual interference caused by the virtual content.

AR technologies are not limited to the display of such obviously-virtual content, however. With proper spatial registration, AR content can be seemingly integrated into the physical environment, going as far as changing the appearance of objects or even people. While this yields exciting perspectives, it also raises ethical concerns which, at the very least, call for a mechanism that lets users remove augmentations on demand. The SceneCtrl system [216] relies on scene reconstruction and texture patches to enable users wearing a Hololens to delete or move a real object in their field of vision. *Annexing Reality* [74] is another system that makes use of actual objects in the user's environment to create tangible proxies for manipulating virtual objects. The system scans the environment for identifying the best match between a physical object and the virtual object to manipulate. The virtual object is used as a texture applied to the physical object, actually modifying its appearance. Lindlbauer *et al.* [111] also demonstrate that, beyond direct texturing, AR can be used to modify the surroundings of a physical object (*e.g.*, to display virtual extensions or cast a shadow). Their system can modify an object's perceived size, color, or shape. HoloFace [100] goes one step further, using face tracking to change people's face. While the paper focuses essentially on the technical realization, such possibilities obviously raise ethical concerns. On this front, Rixen *et al.* [174] study the potential of color alterations applied to clothes or objects, focusing more particularly on interpersonal communication. They identify limits to what users will accept both as the person altering others and as the person being altered.

As graphics rendering and spatial registration technologies improve, it will become increasingly difficult to discern the virtual from the physical. VR technologies can actually be used to implement mixed-reality experiences that go even further, literally creating *"a world in which the physical laws ordinarily governing space, time, mechanics, material properties,* etc. *no longer hold"* [128]. For instance, the Remixed Reality system [112] relies on a set of depth cameras positioned to make a live 3D model of a room, which can then be manipulated before being displayed in the headset. This technique makes it possible to give users the impression that they are flying or walking on walls. Mise-Unseen [120] makes it possible to perform such changes in the environment

in plain sight without users actually noticing. The system relies on eye-tracking to model where users' visual attention is in order to make changes in areas that they are not focusing on.

In such cases where the world *seen* by users does not match the world in which they actually *are*, methods to ensure safety must be implemented. Chaperoning techniques can for instance display a grid visualization of nearby obstacles (such as the HTC Vive's Chaperone system[1]) or indicate a safe trajectory for users to follow [90]. Some research projects also rely on geometrical transformations applied to the displayed world in order to play on users' perception and make them adopt safe walking trajectories [172] and gestures [10]. However, while those techniques can address collision issues, there is a much more diverse type of safety and ethical issues to consider [188] when considering alterations of the real world, as Rixen *et al.* [174] have started investigating. Here again, de-augmentation capabilities might be beneficial to address some of those concerns as we will discuss later.

3.2.3 Augmented Virtuality

Virtual reality fully replaces the real environment with a computer-generated 3D scene. While this helps make the experience immersive, the resulting isolation from the real world can again raise safety concerns, and prevents users from interacting with people and physical objects [124]. This has led researchers to explore the concept of *Augmented Virtuality* where some elements from reality get integrated back into the artificial scene so as to, *e.g.*, avoid collisions with physical obstacles; improve awareness of the social environment; and make it possible to interact with physical things – for instance, to grab a glass of water and drink.

In one of their studies, McGill *et al.* [124] investigated different views onto reality to enable users immersed in VR to interact with physical objects (*e.g.*, type on a keyboard or grab food). They observed that a large window onto the real world conveyed a lesser sense of presence than local views which, for instance, only integrate users' hands in the virtual scene, as in [200]. These results are in line with those of Budhiraja *et al.*'s study [21] in which VR users had to interact with a cup. The technique showing only hand and cup performed better than the one which cut a larger rectangular window onto the real world. Kanamori *et al.* [89] compared three methods to enable users to *perceive* real objects while playing in VR: the HTC Vive's Chaperone system that displays the outline of objects in range of the headset's camera; a visualization that displays a 3D point cloud of objects close to the user; and an visualization that actually displays real objects in the VR scene. The latter method proved the most efficient for a task that required participants to walk and reach a specific physical object such as a cup. Overall, these results suggest that, in the context of VR, seeing some reality is necessary to enable interactions with physical objects and enable safe locomotion in the physical environment, but that the amount of elements brought from the real world should be limited so as not to impact immersion too much.

[1] `https://www.vive.com`

The RealityCheck system [72] recently investigated an application-independent technical solution for the real-time reconstruction of people and objects in the physical space, and their graphical compositing with the virtual environment. This is interesting, as it paves the way for a generic solution to integrate physical elements back in VR. The problem has received less attention in AR than in VR, perhaps because it has been considered less critical so far. The work of Budhiraja *et al.* [22] is an exception. It investigates techniques that combine AR in a headset with a mobile device (as in, *e.g.*, [60, 148, 218]). One of these technique actually *de-augments* the phone when users look down at it, so as to prevent virtual objects displayed in the headset from interfering with the phone's screen.

As AR technologies evolve, they will be used in a variety of daily tasks that involve interactions with physical objects and people. Users' needs in terms of de-augmentation will thus increase. Our contribution is a series of scenarios that illustrate these new needs, and a taxonomy of generic de-augmentation operations. As opposed to VR, which requires the system to decide which elements of reality to show, AR users may still have some perception (even if altered) of the physical environment, giving them agency over what to de-augment, when, and how.

3.3 Illustrative Scenarios

Taking a leap into the future, we now describe various situations in which AR has become pervasive. We illustrate why users would sometimes want to de-augment AR in those situations, and how they could do it.

All scenario figures are using artificial color schemes for the sake of clarity: real-world elements are colored with shades of grey, while AR content is colored mostly with shades of orange. De-augmented areas are outlined with a blue stroke.

3.3.1 Workplace

Kim is working on the design of a hydrogen car with her laptop at her workstation in a large room she shares with co-workers. A physical model of the car's backbone lies in the center of this room. When looking at it through their AR glasses, all workers can see the latest virtual iteration on the car's dashboard, properly aligned with the physical backbone. Kim and her co-workers gather around it to discuss some aspects of the car's interior design.

Like many others, Kim closed her laptop before joining the meeting. She still wants to keep some awareness about incoming messages – both professional and personal – but she does not want those to interfere with the meeting. She has thus configured silent notifications, displayed in AR, world-anchored above her laptop, so that they are not constantly in her way but only show when she glances at the laptop (Figure 3.1-a).

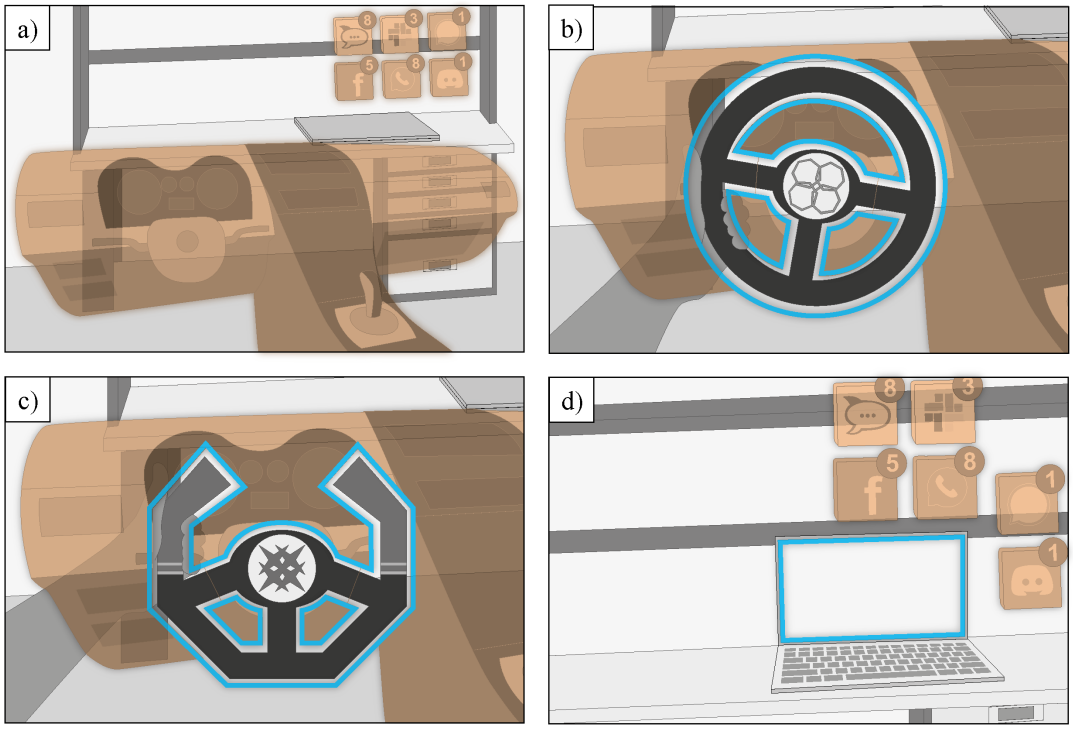


Fig. 3.1: Some de-augmentations from the Workplace scenario. a) Car dashboard and exterior sketches shown to all coworkers in AR, with Kim's desk and laptop closed in the background. Notifications are displayed in the air only to Kim. No de-augmentation is active. b) De-augmenting the first 3D-printed steering wheel, that Kim is holding in her left hand. The wheel can be seen in the context of the dashboard without the latter visually interfering with it. c) Similar de-augmentation applied to the other steering wheel. The de-augmentation's scope matches its shape. d) De-augmenting Kim's laptop screen by shifting some notifications upward.

The team is discussing the optimal placement for the seats and shift lever, considering comfort, aesthetics and mechanical constraints. They also have to make a preliminary choice about two different steering wheel designs, for which they have 3D printed models. As they want to see how each one integrates with the virtual dashboard, Kim triggers a de-augmentation of the wheel based on its bounding polygon. As illustrated in Figure 3.1-b & 3.1-c, this effectively cuts holes in the AR display, resulting in a hybrid representation that seamlessly combines virtual and physical elements.

After an intense discussion, Kim suggests they take a break and go get a coffee. By company policy, AR is not allowed in the cafeteria to ensure a healthy, relaxing break promoting truly-social face-to-face interaction. Management had originally considered asking people to remove their eye-ware in the cafeteria, but quickly realized that there were legitimate reasons to keep wearing it in there too. For instance, Kim has her AR display integrated with her prescription glasses. Jo, one of her co-workers, recently bought the latest technological innovation: true-color AR contact lenses. As any other lenses, those

would be quite impractical to remove just to have a coffee break. Management eventually opted for another solution: people are still allowed to keep their eye-ware, but the devices are automatically instructed to de-augment everything that is not safety-critical when entering the cafeteria's area.

After the break, everyone goes back to their workstation. On leaving the cafeteria, personnel-wide de-augmentation restrictions are removed. Kim gets back to her desk and opens her laptop. The AR notifications floating above it are getting in her way now, however. She points at her laptop and performs a swipe gesture to de-augment them. The system infers that the computer screen is the area to de-augment as the gesture was initiated while pointing at it. It also infers from the context and the explicit nature of this gesture that Kim does not want the AR content that falls in this area to be removed, but rather shifted outside it. As a result, some of the AR notifications get offset upward, as illustrated in Figure 3.1-d.

3.3.2 Home

Alex is an engineer working from home, actually developing a prototype AR application. They just deployed it from their desktop workstation and are in the process of testing it when their phone alarm goes off. This automatically triggers a de-augmentation of the area around the phone, enabling Alex to check it. They see that it is time to prepare dinner for the family. They exit the AR application they were testing in debug mode – resetting their AR eye-ware to normal mode. In this mode, incoming AR notifications appear on the left side of their field of vision. Alex likes this setting as they can check messages while they move around the house.

They start cooking dinner and launch an AR cooking application to get virtual instructions about how to prepare a new dish they have been wanting to try for a while. Meanwhile they receive a notification from their partner that they will be home soon (Figure 3.2-a).

In the kitchen, the induction cooking plates and the smart oven are always de-augmented. This is both for safety and practical reasons, as it enables direct monitoring of the cooking process. Figure 3.2-b illustrates two de-augmentations in the oven area: the AR recipe gets partially hidden where it intersects the oven; and the AR notifications (here the cooking plate countdown at 11 minutes) get shifted downward, right below the oven.

Alex's partner has come home. Dinner will soon be ready. Alex goes upstairs fetch their kids (Figure 3.2-c). One of the kids is playing an AR game that involves catching butterflies that try to hide behind clouds that float throughout the room. As Alex enters, their face gets automatically de-augmented from the child's perspective (Figure 3.2-d). This way they can talk to each other without AR interfering and adversely impacting social interactions [129].

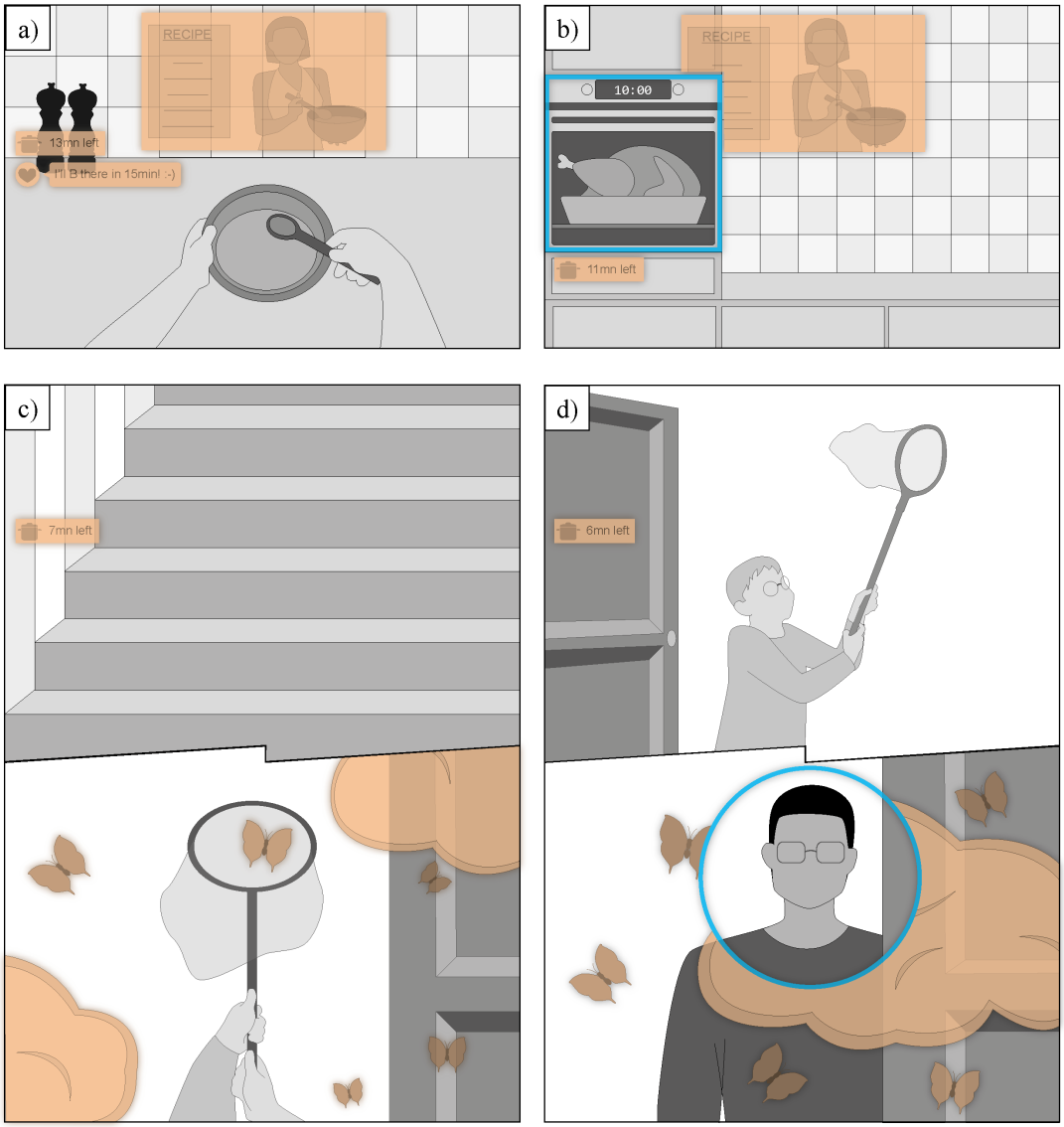


Fig. 3.2: Some de-augmentations from the Home scenario. a) Alex has started cooking in the kitchen. The recipe and notifications appear in AR. No de-augmentation is active. b) The oven is always de-augmented, but the two pieces of AR content get de-augmented differently. The notifications get shifted downard, while the recipe gets cropped. c) Alex is heading upstairs. Meanwhile one of their kids is playing an AR game about catching butterflies that try to hide in the clouds. d) Alex enters the kid's room, and when the kid looks at them, their face gets automatically de-augmented, cropping AR clouds and butterflies.

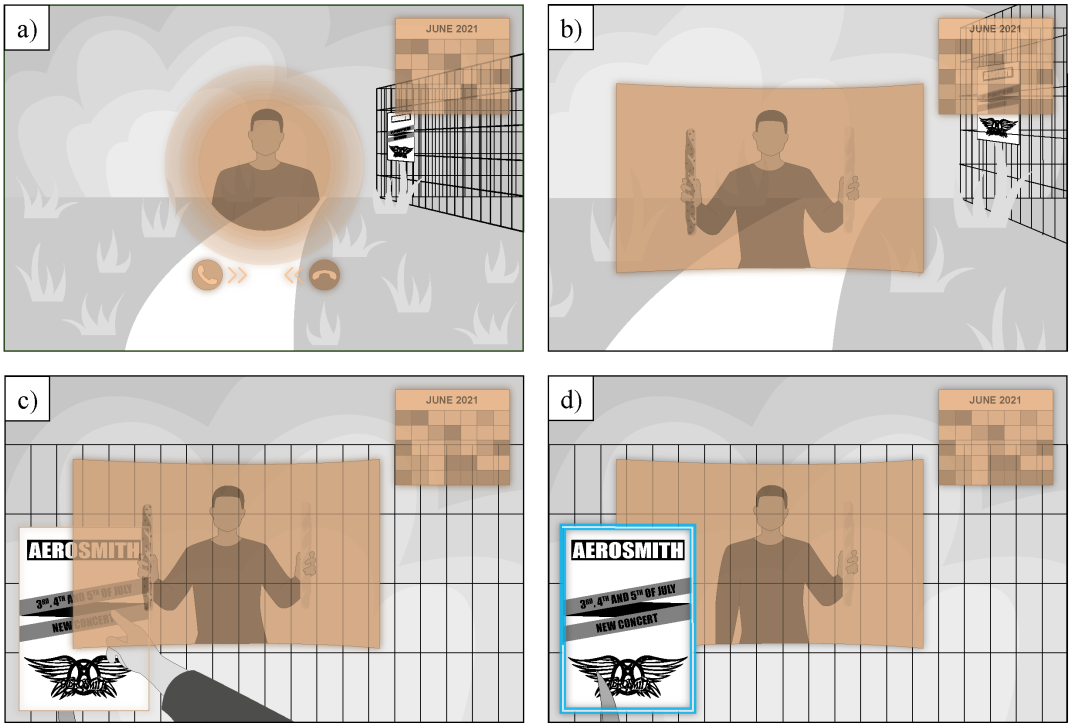


Fig. 3.3: Some de-augmentations from the Outdoors scenario. a) Incoming call from Sam. b) a concert poster catches Jen's attention. c) Jen points at the poster to de-augment it while talking with Sam. d) The poster is now de-augmented. Sam's video feed is cropped.

3.3.3 Outdoors

Jen is taking a stroll in the park. She notices a poster advertising a live music concert when her partner, Sam, calls her (Figure 3.3-a). He is at the bakery and wants to know what type of bread Jen would prefer (Figure 3.3-b). Once the matter is settled, Jen asks Sam to remain on the call, as she thinks the concert is by his favorite band.

Jen got closer to the poster while she was talking with Sam. She can now read the details, but Sam's video feed is in the way. Jen points at the poster (Figure 3.3-c) to de-augment it temporarily with a gesture. She confirms the band and finds out when they will be performing (Figure 3.3-d). She asks Sam whether he would like to go, and suggests one of the three performance dates. Sam seems to remember that they are already invited to a party on that day. Jen checks her calendar in AR, and confirms that it conflicts with their friend's party indeed. Having both her calendar and the concert poster in her field of vision, she finds another date that works. The timer has now expired, automatically canceling the temporary de-augmentation. Jen and Sam keep chatting, with Jen asking Sam to book tickets as soon as he is home.

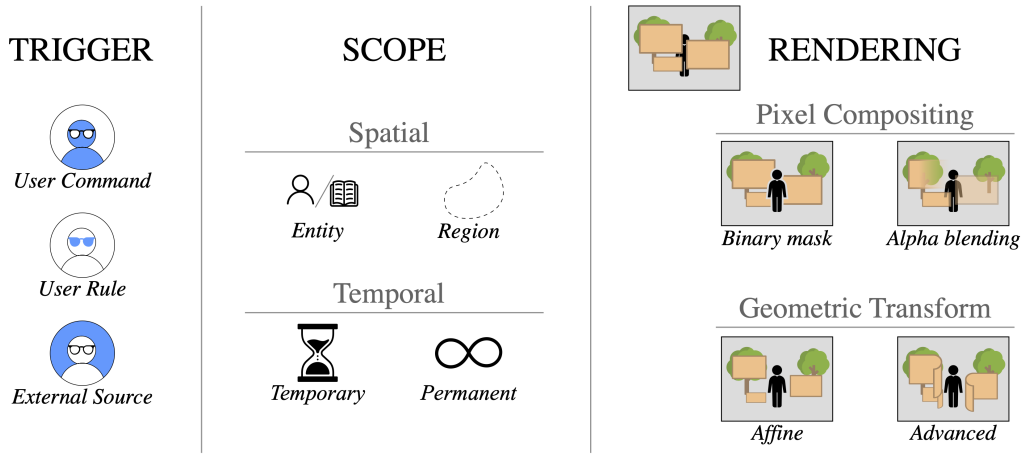


Fig. 3.4: The three characteristics of a visual de-augmentation operation: *Trigger*, *Scope* and *Rendering*.

3.4 De-augmentation Taxonomy

As illustrated in the scenarios above, there is a variety of ways to invoke de-augmentation operations, and a variety of ways to actually de-augment AR content. We already saw that a de-augmentation can be triggered by the user or by an external event, and that it can have a very limited scope or, on the contrary, that it can apply globally. In this section, we first define what a de-augmentation operation is. We then organize these operations into a taxonomy. The overall goal of this taxonomy is to help designers of AR systems think about situations that call for de-augmentation and design techniques to support such operations accordingly.

A de-augmentation is defined as *an operation that alters virtual content in order to improve the user's perception of the real world*. The definition is kept vague on purpose, as the concept of de-augmentation could be relevant not only for visual content, but possibly for modalities such as audio and haptics as well. As we focus here on visual de-augmentations, we can also give a more specific definition of a *visual* de-augmentation: *an operation that alters virtual content in the user's field of vision in order to improve their visual perception of the real world*.

We decompose a visual de-augmentation operation into three components:

- what *triggers* the de-augmentation;
- what is its *scope*, *i.e.*, what part of the real-world gets de-augmented and for how long;
- how is the de-augmentation achieved, *i.e.*, how is the virtual content's *rendering* altered.

3.4.1 De-augmentation Triggers

AR de-augmentations happen in response to a trigger. We distinguish three main types of triggers. The first two are specified by users themselves: either on-demand by invoking a *User Command*; or automatically, the system triggering *User Rules* that match the current situation. The last type of trigger is specified by an *External Source* independent from the user. These can be rules hardcoded for safety or privacy reasons, or imposed by authorities who have jurisdiction in the current context.

User Command

A de-augmentation can be triggered with a ***User Command*** invoked explicitly. Such commands can involve any input modality. In the workplace scenario, Kim performs a swipe gesture to de-augment her laptop's screen. In the outdoors scenario, Jen points at the poster advertising the concert. While these examples involve hand movements (gestures), any other interaction technique that works well with AR technologies can be considered, such as voice or gaze. One could for instance envision giving users some control on the level of (de-)augmentation based on how they look through their glasses. Just the same as the wearer of prescription glasses equipped with progressive lenses does not get the same amount of correction depending on whether they look through the lower, middle, or upper part of their eye-ware, the wearer of AR glasses could have some control on virtual content (de-)augmentation. This would only give them a relatively basic level control, however.

Depending on the other characteristics of the de-augmentation to be specified, more expressive input might be required, for instance to define a specific scope or a specific way to de-augment the virtual content. Simple deictic pointing combined with computer vision techniques can for instance be used to de-augment specific objects. Freeform gesture tracking enables users to delineate arbitrarily-shaped spatial scopes such as, *e.g.*, a sketch on a whiteboard or an irregular area on the floor. We can even think of using other modalities such as a handheld device (*e.g.*, a smartphone) or tangibles. For example, drawing inspiration from Butz *et al.* [25], a tangible could play the role of a lamp that de-augments the region it illuminates. Tangible posts could also be used to delimit a large area, in the spirit of the ShareSpace system [215].

Finer control on the de-augmentation's properties can also be provided with traditional UI widgets in AR, for instance to adjust specific rendering parameters such as the transparency of de-augmented areas. However, we can expect that advanced and time-consuming de-augmentation specifications would rarely occur, elaborate configurations being better managed as automatically-triggered rules when possible.

User Rules

Contrary to user commands, ***User Rules*** do not require explicit actions from the user. They are triggered by the AR system automatically, whenever they apply to the current context. In the workplace scenario, a de-augmentation is triggered to let Kim see the steering wheel because she has configured the system so that it will de-augment the area of her field of vision that corresponds to any object she holds in her hands. This is a general rule, that she applies in any context, that would also let her see the glass or cup she intends to drink from [124].

Such user rules remove the need to take explicit action to trigger a de-augmention, but as illustrated just above, they also remove the need to specify a scope for the de-augmentation. Another example consists of de-augmenting people's face in the household, as Alex and their family have done in the Home scenario. More elaborate examples include Alex's phone getting de-augmented because an alarm went off; or a window in their house getting de-augmented whenever they stand less than one meter from it as they enjoy relaxing by looking at the stunning view over the mountains they get from there.

We can also envision rules that would fire at regular intervals. For instance, previous work has pointed at the actual risk associated with too much AR assistance, that could make everyday life *too facile* [97], and make dependency on technology too strong [176]. People who want to self-regulate their use of AR or just make sure that they stay in touch with (totally-unaltered) reality could configure a rule to de-augment everything for a few seconds every five minutes.

External Source

While many rules will have been configured, or at least activated, by the users themselves, we can also envision that some rules would be imposed by ***External Sources***. These de-augmentations would be beyond users' individual control. As mentioned earlier, authorities having jurisdiction in a given area might enforce a no- or limited-AR policy for safety, privacy, or security reasons. Buildings such as police stations or any type of restricted area where sensitive activities are performed are obvious candidates, but trafic-heavy streets (at least when about to cross them) might also be considered as such. The workplace scenario illustrates another sort of trigger from an external source, where AR content gets de-augmented when entering the cafeteria by company policy, except for safety-critical information.

3.4.2 De-augmentation Scope

The second defining characteristic of a de-augmentation is its scope, which can actually be broken down in two (Figure 3.4): the spatial scope, and the temporal scope. The spatial scope can be defined by physical entities (people, objects) no matter where they are; or it can be defined as a specific region, no matter what entities it contains. The temporal scope is orthogonal to the spatial scope. It can be either permanent or temporary.

Spatial Scope: Entity

De-augmenting an entity means de-augmenting virtual content that visually interferes with the perception of this entity, regardless of its actual location in the physical world. Typical examples include people (face or entire body); physical objects; as well as physical displays such as computer screens or smartphones. An entity's location can change over time, implying that the system must be able to track it in the user's field of vision in order to dynamically adapt the de-augmented area.

Alex's smartphone and face (home scenario), or the concert poster (outdoors scenario) are examples of de-augmentations whose spatial scope is an entity. A scope can consist of multiple entities. In collaborative AR work contexts [15] such as the one described in our workplace scenario, multiple entities of different nature get de-augmented simultaneously: users share the same overall virtual scene, but de-augment not only physical objects (such as the 3D-printed steering wheel) but their collaborators' face as well to preserve the directness and naturalness of face-to-face interpersonal communication [129].

Spatial Scope: Region

De-augmenting a region means de-augmenting virtual content that visually interferes with the perception of a region in the user's field of vision, regardless of what physical entities it contains. A region's location does not change over time but its contents can change, without impacting the de-augmentation. A region can be either *world-anchored* or *FoV-anchored*. A world-anchored region is fixed in the physical world, regardless of the user's location and viewing direction. A FoV-anchored region is defined relative to the user's field of vision, as if it were attached to the eye-ware.

The cafeteria in the workplace scenario is an example of world-anchored-region scope. The oven and cooking plate de-augmentations in the home scenario can actually be considered as instances of world-anchored-region scopes. On a conceptual level, these are entity de-augmentations. But because those entities are stationary for most practical purposes, handling them as fixed regions removes the need for dynamic object tracking. FoV-anchored-region scopes are perhaps less common. They can be useful to guarantee the absence of visual interference in part of the user's field of vision regardless of any other consideration. They could also serve purposes such as before/after comparisons: with half of their field of vision showing virtual content but not the other, users could compare, *e.g.*, actual *vs.* virtual clothes or face with/without makeup.

Temporal Scope: Temporary

A de-augmentation whose temporal scope is temporary will only last some pre-defined duration, after which it will automatically get cancelled, the virtual content appearing back.

Jen's de-augmentation of the concert poster in the outdoors scenario is temporary, as she only needs to look at it for a few seconds. Having to cancel the de-augmentation explicitly would be cumbersome. Similarly, Alex's smartphone could get de-augmented for three or four seconds when notifications pop up. Temporary de-augmentations can also be useful to lower the *level of engagement* [164], as an AR game player might want to in order to eat a snack for instance.

Temporal Scope: Permanent

A de-augmentation whose temporal scope is permanent ∞ lasts until it is explicitly cancelled, either by the user or by an outside event that results in a change of context. For instance, the rule that triggered the de-augmentation might no longer apply.

The de-augmentation in the cafeteria from the workplace scenario, the oven and cooking plates in the home scenario as well as Alex's face are all examples of permanent de-augmentations. Even if they do not necessarily last very long, the steering wheel and the laptop screen de-augmentations in the workplace scenario are considered permanent as they are active as long as those entities remain in the user's field of vision.

3.4.3 De-augmentation Rendering

Once triggered in a specific scope, a de-augmentation is achieved by altering the virtual content that falls within that scope. The simplest de-augmentation consists of not showing anything in the corresponding area of the user's field of vision, as if cutting a hole in the virtual content. But alternative rendering strategies can also be considered (Figure 3.4).

Pixel Compositing

Cutting holes in a layer to reveal hidden objects has been experimented in many different situations, including window management [199], map comparisons [114], or layered 3D models [161]. Drilling holes in the AR layer can be as straightforward as applying a binary image mask  to the rasterized virtual content right before display in the eye-ware. But this is just one particular option in the broader category of pixel-based image compositing strategies. Pixels that fall within the de-augmentation's scope or in its surroundings can have their alpha channel adjusted  without necessarily making them fully transparent. Content-aware techniques [81, 207] are also quite interesting in our context: they could be used to adjust the level of transparency of individual pixels based on the comparative importance of the virtual and real content in a given area. Most of the de-augmentations in our three scenarios are examples of compositing-based de-augmentations.

Geometric Transform

Virtual content can also be removed from a given scope in the user's field of vision by applying transformations to elements in the AR layer. One of the simplest transformations consists of applying affine transforms  to individual elements that intersect the de-augmentation's scope. With a combination of translations and downscaling operations, elements from the AR layer can be prevented from visually interfering with elements to be de-augmented. Notifications in both the workplace and home scenarios are examples of such de-augmentations.

Beyond affine transforms, more advanced transforms  that distort the geometry of elements in the AR layer can be considered. For example, content-aware focus+context techniques [160] could be adapted to distort elements in the virtual layer so as to better integrate them with the de-augmentation scope's shape. Desktop techniques can also be a source of inspiration, based on the metaphor of rolling or folding layers as if they were paper [32, 42, 45].

3.5 Proof-of-concept Prototype

Many of the situations described in our scenarios take a more or less distant leap in the future, making the assumption that advancements in eye-ware display technology has led to the broad diffusion of AR. It is still interesting to assess the technical feasibility of de-augmentation with current AR devices as the principles should essentially remain the same. We developed a proof-of-concept prototype to illustrate and test several of the de-augmentations from our taxonomy. This prototype runs on a Microsoft HoloLens 2 (OS Build `10.0.19041.1131`), and was developed with Unity `2019.4.11f1` and the Microsoft Mixed Reality Toolkit (MRTK `2.5.4`).

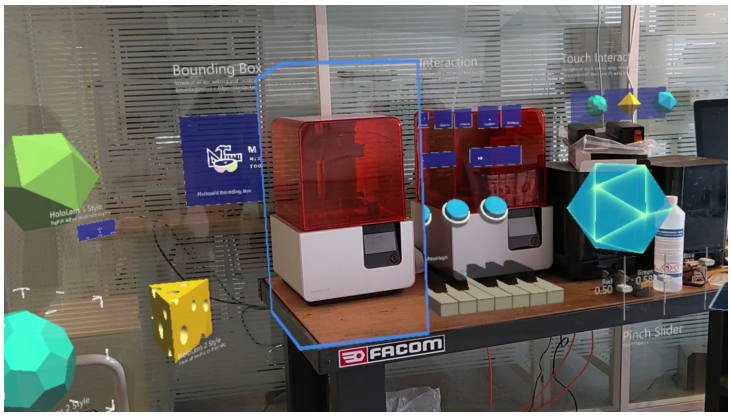
The prototype, illustrated in Figure 3.5 and demonstrated in the companion video, lets users create multiple de-augmentations by invoking a menu. They first specify the type of spatial scope: *world-anchored* or *FoV-anchored*. They can then select among multiple options to delimit it.

The first technique to define a world-anchored scope relies on MRTK's `SceneUnderstanding` feature to detect surfaces in the physical environment. Users point at the surface[2] they want to de-augment and perform an air tap to actually select it. Other techniques involve drawing primitives such as lasso or rectangular selection tools. The interaction sequence differs depending on the drawing tool considered, seeking to make interaction as simple as possible. With the lasso tool, users can draw arbitrary shapes; the system will then sample points along the drawn curve and reconstruct a 3D surface from them. With other tools, users only have to specify two points in 3D space. The system will constrain the bounding box of the selected geometric primitive (cuboid, ellipsoid) so that two of its faces remain parallel to the user's sagittal plane. Regardless of how these 3D scopes are specified, pixels that fall in their projection in the user's field of vision will be de-augmented, as illustrated in Figure 3.5 b-c.

[2] The system identifies the surface closest to the head-hand ray.

a)

b)



c)

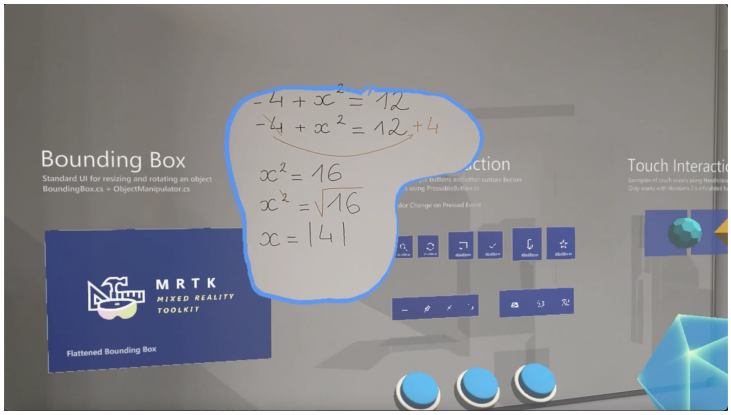


Fig. 3.5: De-augmentions performed with our prototype, performed on the `HandInteractionExamples` scene from MRTK. a) User invoking a hand menu with their hand flat in the Hololens' viewport. The user can choose how to delimit the scope of a de-augmentation to be manually created. b) Example of a world-anchored cuboid de-augmentation to reveal a 3D printer. c) Example of a world-anchored freeform flat de-augmentation to reveal part of a whiteboard.

Users define FoV-anchored scopes by delimiting an area on a 2D surface coplanar to the headset's screen. Usual drawing tools are again used for this purpose, including lasso, rectangle and ellipse. An additional *eraser* tool lets users delimit a scope by holding one hand flat in front of them and moving it, as if they were wiping the virtual plane. As in the world-anchored case, pixels that fall in those areas will be de-augmented. The areas remain tied to the headset, however: the de-augmentations remain fixed in the user's field of vision.

Beyond manually-specified de-augmentations, our prototype also supports some simple *User Rules* aimed at demonstrating automatically-triggered de-augmentations. These rules de-augment faces, the wearer's hands, and objects that have been tagged. Hand tracking is readily available in the Hololens. Face recognition relies on OpenCV `4.1.2` and HoloLens2ForCV [202]. Objects are tracked by attaching Vuforia tags to them and declaring them in the system.

Our prototype can render de-augmentations based on pixel compositing. Pixels that fall in a scope are made fully transparent – thus revealing the physical world – using `ColorMask` shaders, one for each de-augmented area in the user's field of vision. The shader removes any color in the area and forces the shape to be rendered first in the queue, right in front of the near plane, so that the de-augmentation mask will apply to all objects behind it in the virtual scene.

3.6 Discussion

Technological advances let us foresee a future where AR eye-ware has become so lightweight and inconspicuous that reality can be augmented most of the time, if not all the time. The purpose of this chapter is not to advocate for such a future, but rather to highlight the need to give users some agency over virtual content, enabling them to *de-augment* augmented reality. While toggling the full AR display off is a special case of de-augmentation, we are rather interested in locally-bounded de-augmentations, that remove virtual content in some parts of the users' field of vision, where it actually interferes with their perception of reality only.

De-augmentation, as a concept, is different from hybrid representations in VR that bring elements from the real world back into the user's virtual world. VR completely occludes the real world. Research on hybrid representations has consequently focused on integrating elements from the users' physical surroundings based on the assumption that the system would be in charge of detecting what elements from the real world should be integrated into the virtual scene. Because AR superimposes content on a virtual, transparent layer in the user's field of vision, it does not completely occlude the real world. It is thus inherently hybrid, and raises different challenges. In AR, users can still perceive the world around them. Even in a crowded field of vision, they will see the real world through the virtual content – albeit degraded. While this content can impede users' perception of their physical surroundings – one of the very reasons for enabling de-augmentations – this ability to see through the virtual layer is key to giving users

agency over how to mitigate visual interference. The challenge lies in providing users with easy means to de-augment, both manually (interaction techniques) and automatically (context-dependent rules). Another challenge comes from the versatility of AR, which has different use cases than VR. Perhaps a broader range of them, but most importantly, many that involve mobile contexts of use which are much more unpredictable than the typical VR room. Virtual content will be "in the user's way" in a variety of situations, all of which cannot possibly be anticipated by AR system designers and engineers.

In this chapter, we set forth a series of scenarios that illustrate when de-augmentations might be relevant. Examples developed in these scenarios included: seeing faces when talking to people while staying immersed in a dense AR environment (kid playing a game in the Home scenario); combining information from the physical and the virtual world while on a call (concert poster, calendar and remote video feed in the Outdoors scenario); guaranteeing the visibility of safety-critical appliances (oven and cooking plates in the Home scenario); immersing physical objects in their virtual surroundings (steering wheel inside the car in the Workplace scenario); enforcing social rules in public spaces (cafeteria in the Workplace scenario).

More scenarios calling for de-augmentation capabilities exist. These are the ones that informed our taxonomy of AR de-augmentations. This taxonomy represents the main contribution of this work. It aims at providing a common framework to talk about de-augmentation. It also aims at helping designers integrate de-augmentation in their AR experiences. The taxonomy characterizes a de-augmentation based on three elements: who or what triggers it; what is its scope (both spatial and temporal); and how is the de-augmentation achieved visually. The goal is not only to organize de-augmentation techniques, but to inspire future designs as well. The taxonomy itself will likely need to be expanded, or at least refined, as new AR usage scenarios emerge. For instance, the *External Source* trigger category is fairly broad. There might also be some specific triggers that do not fit nicely in one of the categories. Let us consider the case where a de-augmentation should be triggered when somebody waves at an AR user. The trigger is essentially from an *External Source*, but also requires that a *User Rule* exist on the AR user's side to authorize such de-augmentations initiated by third parties. This might be acceptable when the de-augmentation originates from relatives, friends and colleagues. For example, the parent in the Home scenario might have had to actually wave at their kid instead of being automatically de-augmented, if only to prevent unwanted de-augmentations when just passing by. But such third-party triggers might not be acceptable when they originate from mere acquaintances or bystanders.

This raises the broader question of which external sources might have control over a user's AR de-augmentations, when, and how. Yang *et al.* [215] investigated how to enable both VR and non-VR users to share the same physical space. Non-VR users could "protect" themselves from VR users wearing an HMD by positioning virtual shields (visible in VR) in the room. The intention was primarily to avoid physical collisions,

but this work draws attention on the issue of enabling both XR and non-XR users to be in the same space. In particular, what sort of control should XR and non-XR users have over the virtual content? What sort of control should authorities have in areas under their jurisdiction? The notion of sensitive areas with restricted physical access, or where taking pictures is not allowed, could be extended to limit the use of AR. Safety-related de-augmentations such as automatically removing virtual content interfering with emergency exits would seem to make sense, but what if the user had special needs that made the virtual content key to their safety? Could these external triggers be overridden by the user's system? These questions have safety and societal dimensions (privacy, ethics) that obviously go well beyond the design of a taxonomy of de-augmentations. But the concept of de-augmentation seems relevant to the broader matter of AR-related laws and regulations. Keeping in mind that in many scenarios a full AR deactivation will be neither desirable nor necessary, and drawing a parallel with accept *vs.* block lists: while some cases are best handled by preventing an augmentation in the first place, other cases are best handled by de-augmenting part of what has been augmented in the first place.

3.7 Future Work

Beyond safety and societal considerations and the corresponding needs in terms of regulations and protocols, there are also more practical questions that pertain more directly to interaction design. One of them relates to the cost of specification and trigger of de-augmentations. *User rules* activate automatically based on the user's context, but necessarily have an initial setup cost. De-augmentations based on *user commands* are specified manually on the spot. Their creation cost is not necessarily very high, as they should be fairly simple (typically applied directly to a given region or entity). But manually creating a de-augmentation will require users to interrupt their primary task, which could be tedious and frustrating.

Streamlining the de-augmentation process from an interaction design perspective is a key issue. The prototype framework introduced in Section 3.5 features examples of both *user commands* and predefined *user rules*. Implemented with current off-the-shelf technology, this prototype primarily aims at demonstrating that de-augmentation is not just a concept, but an actual possibility already. The interaction techniques developed to manually create de-augmentations are very crude and only enable simple de-augmentations. Much work remains to be done to lower the cost of de-augmenting with *user commands*. This interaction design work, however, will be heavily dependent on the input and tracking capabilities of future AR devices. The specification of *user rules* is maybe less dependent on those capabilities, at least in terms of input modalities. But implementing effective rule systems will require multiple sensors in order to infer the user's context and activate de-augmentations accordingly.

Contextual information could actually be used to assist users in the specification of the rules themselves, in order to minimize the setup cost. For instance, the system could learn from users' past actions: if a user frequently de-augments windows in their house

when they are less than one meter from them, the system could suggest adding a rule to automatically de-augment any window within this perimeter. The system might also learn to apply different rendering strategies to de-augment particular pieces of virtual content, for example making notifications semi-transparent instead of masking them based on the inferred user's degree-of-interest in this type of content. Contextual information could also help create de-augmentations manually. As mentioned earlier, our prototype lets users select surfaces detected by the HoloLens 2 as de-augmentation scopes. More elaborate object detection and scene understanding capabilities would help streamline the interactive de-augmentation process. This would mitigate the earlier-mentioned interruptions in the user's primary task, which is key to making de-augmentation an effective solution for managing visual interferences caused by virtual content.

Taking a more prospective look at future uses of de-augmentation, the concept might find interesting applications as we get closer to the ultimate goal of *"generat[ing] virtual objects that are so realistic that they are virtually indistinguishable from the real environment* [11], possibly helping locally differentiate the virtual from the real. Another direction for future work will be to investigate the concept of de-augmentation applied to other AR modalities such as sound and haptics. Technological advances (audio spatialization, noise-cancelling techniques and ultrasound-based haptics just to mention a few) yield exciting opportunities to manipulate these signals. Here again, and as per the definition of de-augmentation given in Section 3.4, the idea would rather be to investigate the feasibility and benefits of *partially* de-augmenting the sounds and haptic feedback generated by the AR system – as opposed to fully deactivating them – so as to improve users' perception of the real world across more senses than vision alone.

3.8 Conclusion

Technology advance promises a future where Augmented Reality content can be blended with the real environment. This definitely affects users' perception which can be tricky, especially for security and social concerns. In this chapter we have presented the concept of *de-augmenting* AR as a way for users to choose what they want their focus to be on, being constantly in control of what they perceive. We have first illustrated the concept of *de-augmenting* with three scenarios that illustrate how *de-augmention* could appear in three different daily contexts: 1) at work, 2) outdoor, and 3) at home. We introduced a taxonomy of *de-augmentions* and thus, delineating the main components in the design of *de-augmenting* techniques, which are: 1) the triggering, 2) the scope, and 3) the rendering.

We implemented an actual prototype combining several tools for de-augmentation. Our prototype shows the feasibility of letting users decide how to organize their field of view, deciding on what the elements of interest are in what they view. While we argue that AR eyewear could be integrated into users' daily lives,we also point at the risk for users to lose control over their perception of the real world with the advance of AR technology. There is a real need for involving users in the blending process between both the real and the virtual environments.

Part III

CONCLUSION AND PERSPECTIVES

Supported by the HCI literature as well as my own research, I strongly argue that Augmented Reality has a great potential of being adopted by users in different daily contexts. However, we are not there yet. As discussed in this manuscript, it is not only related to the maturity of the technology, it is also because of usability and utilisability issues [98]. Such issues range from low-level ones such as the fatigue that is inherent to mid-air interaction, to high-level ones such as the acceptability regarding safety and social aspects. This thesis is an investigation of how AR could be reconsidered to be adapted to daily situations. We identified three specific research questions and reported on research projects to answer them. Each of these questions relates to one aspect of the adoption at large of AR technology: 1) users' motor performance when interacting with AR ontent; 2) integration of AR technology in users' digital environment; 3) users' agency regarding where AR content should be displayed. This thesis' contributions are in terms of interaction design, empirical results and conceptual frameworks. This chapter gets back to each individual research question, provides a summary of contributions and opens questions for future work.

RQ_1 How to reconsider interaction with wearable AR to better support long-lasting tasks?

To answer this research question, we present the *ARPads* project, studying the performance of mid-air indirect input for Augmented Reality as a means to reduce fatigue when interacting with AR content. We characterize a design space for such interactions, based on the location and orientation of those pads relative to the user's body. We report on a study, comparing six of these *ARPads* to their state-of-art counterparts for continuous and discrete interactions. Our study shows that some ARpads can reduce users' fatigue while reaching time and precision performances that are similar to or better than the state-of-art counterparts. We also provide guidelines for developers to implement such *ARPads*.

Our current implementation of *ARPads* relies on absolute input. This means that the absolute position of the user's hand whithin the *ARPad* is mapped to the same absolute location whithin the cursor's space, whith a Control-Display ratio that is the same over the whole surface. Future investigations could explore the use of a relative mapping between the user's hand movements and the cursor displayed in the AR screen. While absolute input makes it fast to reach a specific point in the interactive area [123], relative input techniques could bring new advantages. In particular, they make it possible to change the Control-Display ratio, which is often useful to perform precise movements [142]. The Control-Display ratio can also be defined as a function of the UI semantics [16] in order to facilitate interaction with some specific UI elements. Finally, relative mapping can also take advantage of transfer functions that consider precision as a function of movement speed [29]. A promising direction could be to consider a combination of both absolute and relative input such as *HypridPointing* [53], which could make *ARPads* benefit from both relative and absolute mapping advantages. Such improvements would open up the applicability of *ARPads* to larger interactive spaces than the medium-sized AR window that we tested in our study.

Another interesting perspective for *ARPads* relates to their shape. Our experiment only considers 2D rectangle pads, with an aspect ratio that is identical to the virtual window displayed in AR. In future work, we would like to explore 3D indirect mid-air input. The 3D AR volumes could either have a basic, cuboid shape or even a more complex one that fits the geometry of the specific object to manipulate. In both cases, this future work would likely require more design work regarding feedback about the shape and the location of the virtual interactive volume. In our project, we displayed a transparent, red layer over the user's field of view when the user's hand leaves the *ARPad*'s virtual plane . This solution might be not efficient enough to help users to have a clear representation of the 3D AR volume boundaries. To help users get a clear mental representation of the control volume, we could think of using *e.g.* haptic [115] or audio feedbacks to better convey the boundaries of this volume.

RQ_2 Can we provide guidelines on how a wearable AR device can enhance a handheld device?

We address this research question by studying potential enhancements to phone-centric interactions. We report on a workshop with users and describe a design space to organize users' suggestions for enhancing phone interactions with AR. This design space features two dimensions that relate to the location and the function of the AR content with respect to the phone's content. After prototyping some of the examples to showcase promising cells in this design space, we focus on the AR enhancement of traditional smartphone-based widgets. We run a study to define how the user interface could be distributed between a handheld device and AR. This study shows that handheld devices could benefit from being AR-enhanced, be it for users' performance, or for freeing up the phone's screen estate.

In our project, we only focus on how a user can interact with an AR-enhanced interface for smartphones, without considering the prior step of configuring this AR-enhanced interface. Such customization could be either automatically generated based on users' past interactions, or entirely controlled by users themselves. Automatically generating an AR-enhanced interface for smartphones could rely on dedicated algorithms like SUPPLE [56]. However, prior research suggested that there is a need for keeping the user in control of the adaptation process [209, 49]. An interesting approach could be hybrid with a system like Skillometers [117] that would provide means for users to reflect on their activity and identify what gain they could get from some customization. Studying how users could decide on whether they want to offload a widget or not is an interesting avenue for future work. Some researchers have already worked on such adaptable user interfaces [195] on desktop, but this needs to be explored in the context of AR+smartphone combination.

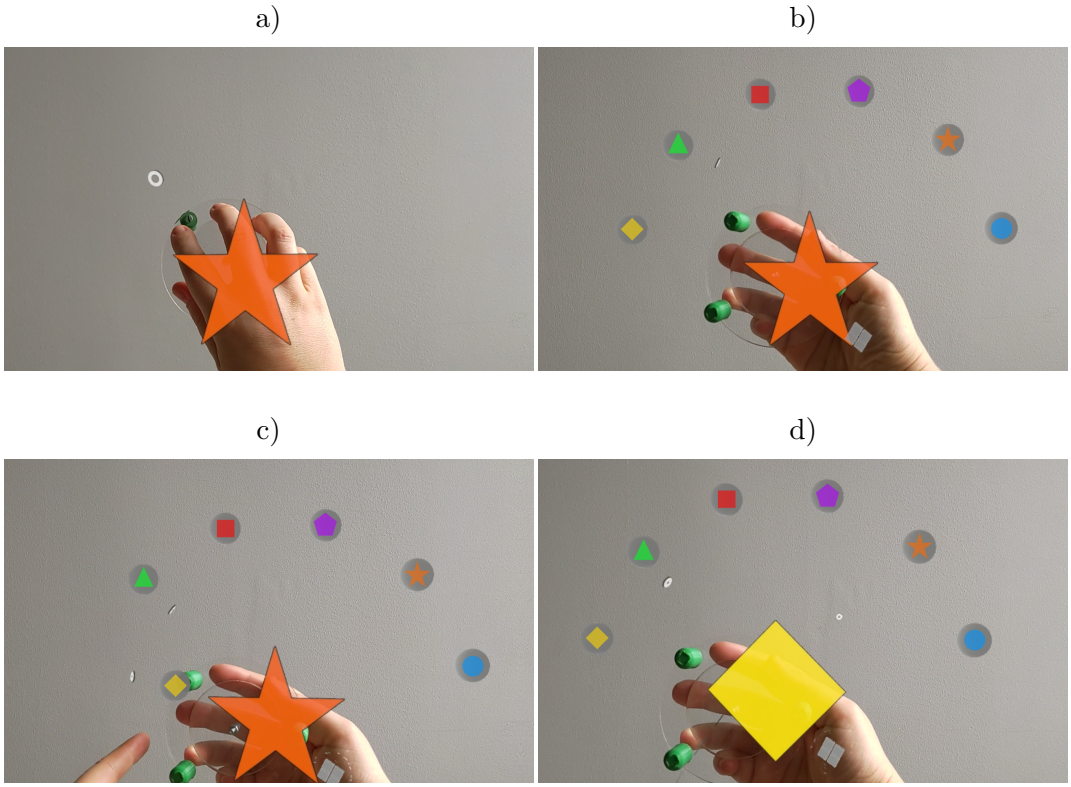


Fig. 3.6: Initial prototype of AR-enhancement for tangible tokens. a) The user has already selected an orange star as the appearance of the physical, transparent WallToken [37]. b) Then, the user turns their hand to display the menu that proposes other possible appearances of the token. c) The user moves one menu item to the physical token by sliding their finger toward it. d) The appearance of the token is updated.

We would also like to continue working on AR combination with interaction modalities other than touch-only. In particular, we have started to work on the combination of AR with tangible controllers such as TouchTokens [138] or WallTokens [37]. Tokens could benefit from this combination as they are physical objects whose shape cannot be changed on the fly [7, 79]. Using a standard physical placeholder for tokens and changing the way they look in AR could make them easy to customize. More than just preventing users from having to manipulate too many tokens and interaction designers from creating too many token instances, this solution could preserve the advantages of physicality like haptic feedback while overcoming its drawbacks like resistance to change. We have already implemented an initial prototype for users to define the appearance of one token using AR (illustrated in Figure 3.6). Users can simply turn their hand holding the token to invoke a menu that proposes different visual appearances (and thus functions) that the token can take. The user can then customize the token by sliding their finger from the desired menu item to the physical token.

RQ_3 How can users be in control of what Augmented Reality tells them?

To provide a first answer to this question, we present the concept of *de-augmenting* Augmented Reality. We describe a taxonomy of *de-augmentations* regarding their triggering, scope, and rendering. We illustrate our taxonomy through three scenarios in order to demonstrate how this concept would help integrate AR in different daily contexts. We also contribute a prototype that shows the technical feasibility of implementing *de-augmentation* operations that users have control over.

This project is a preliminary investigation of letting users control what falls in their field of vision when using AR. But AR can be more than visual only. Multimodal interfaces have been studied for a long time [147, 183, 102]. They can take advantages of users' multiple sensory channels to enhance user experience. Researchers have already explored a wide range of augmented modalities [147, 183, 102] that include haptic [154], smell [116], taste [204], sound [189] or temperature [170]. Envisioning a future where AR is multisensory, we should enlarge the scope of applications for our *de-augmentation* concept. As a future project, we could think of letting users *de-augment* one or several AR feedback channels. More than just helping the user be in control of their various perceptual channels, this could make AR systems more accessible for people with disabilities. As an example, people with autism spectrum disorders can be especially sensitive to loudness [94], and people with headaches can encounter osmophobia [187]. Such channels' *de-augmentations* could be tailored to suit these users' specific needs.

Part IV

ANNEXES

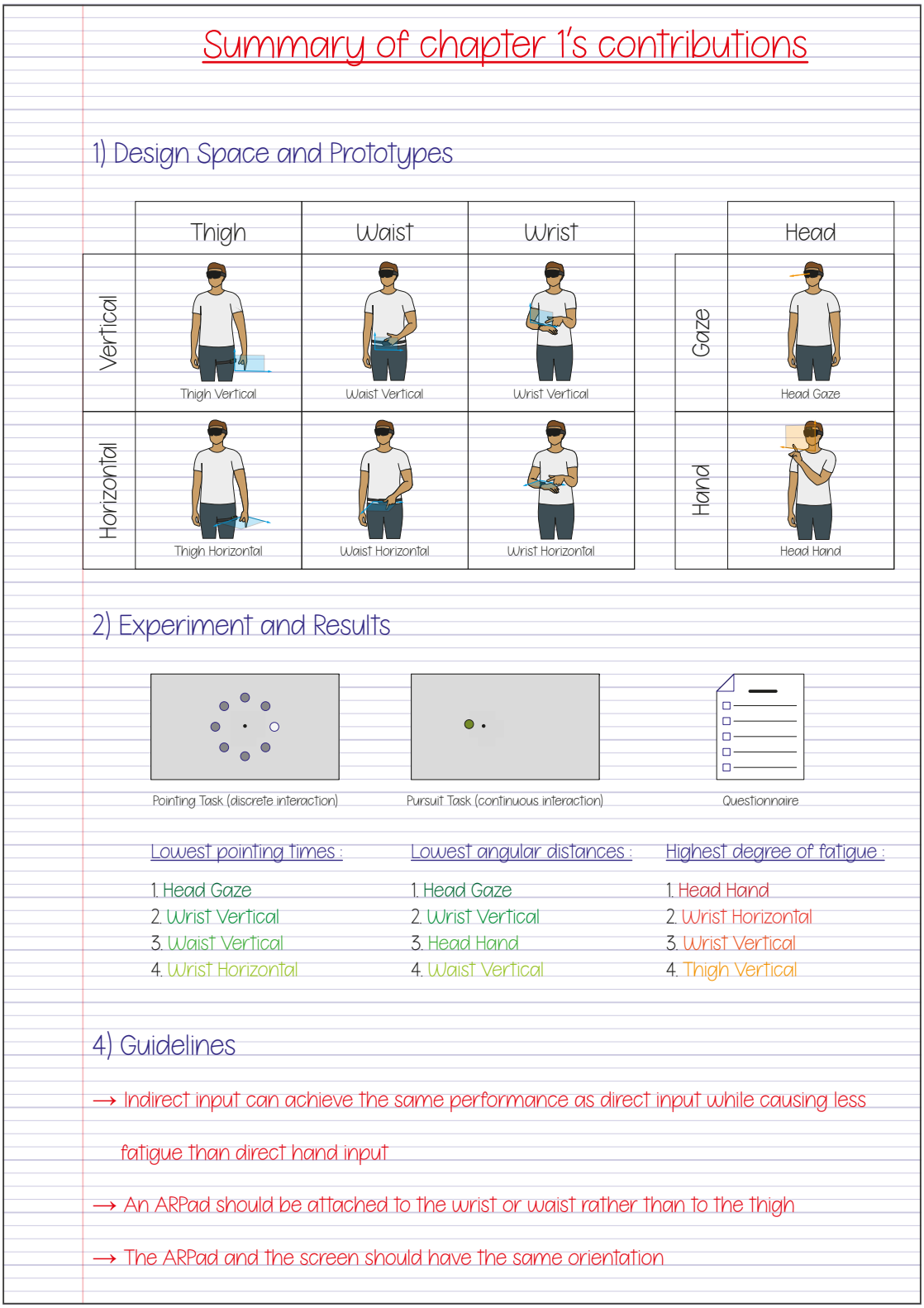


Fig. 3.7: Summary of chapter 1's contributions.

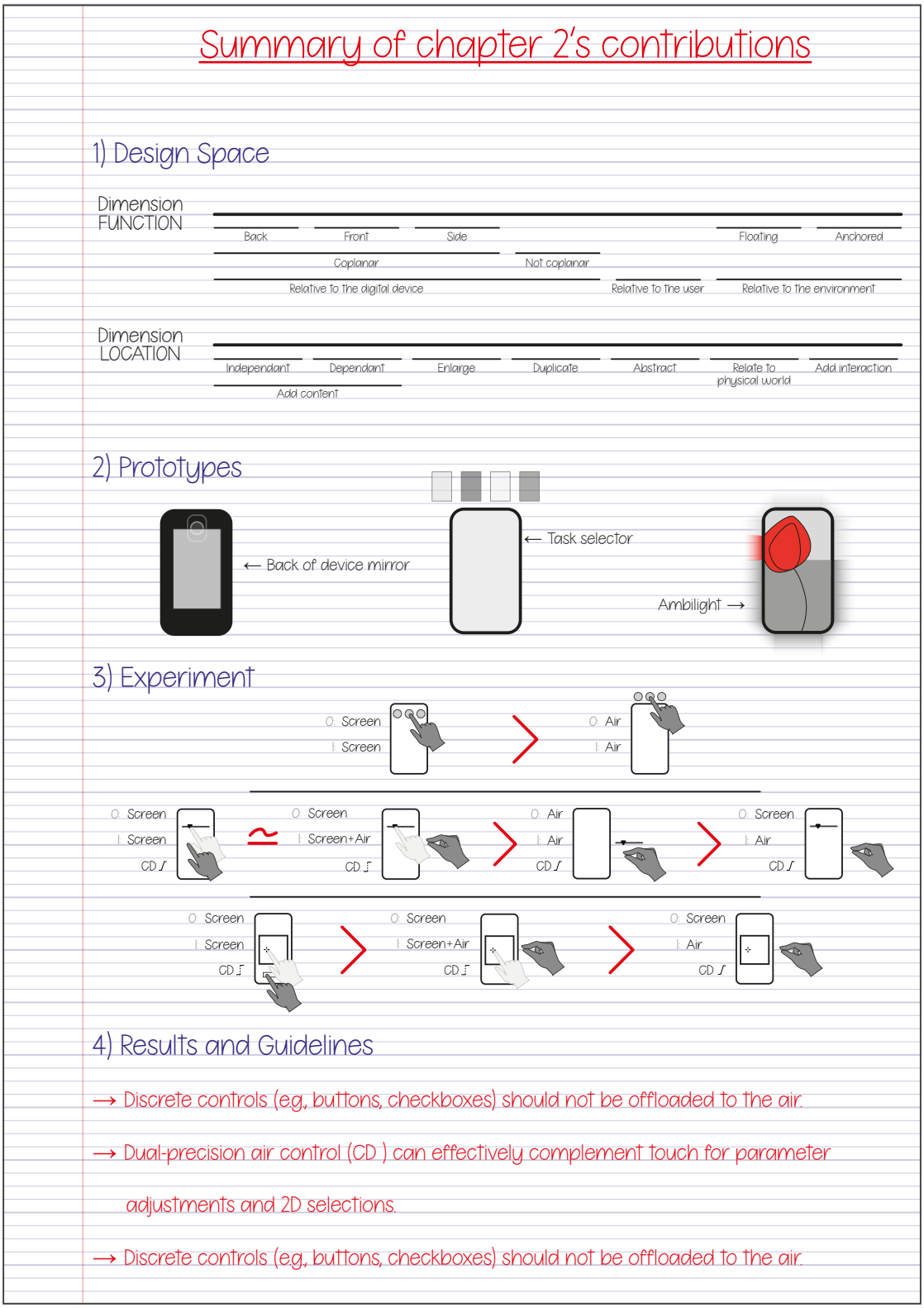


Fig. 3.8: Summary of chapter 2's contributions.

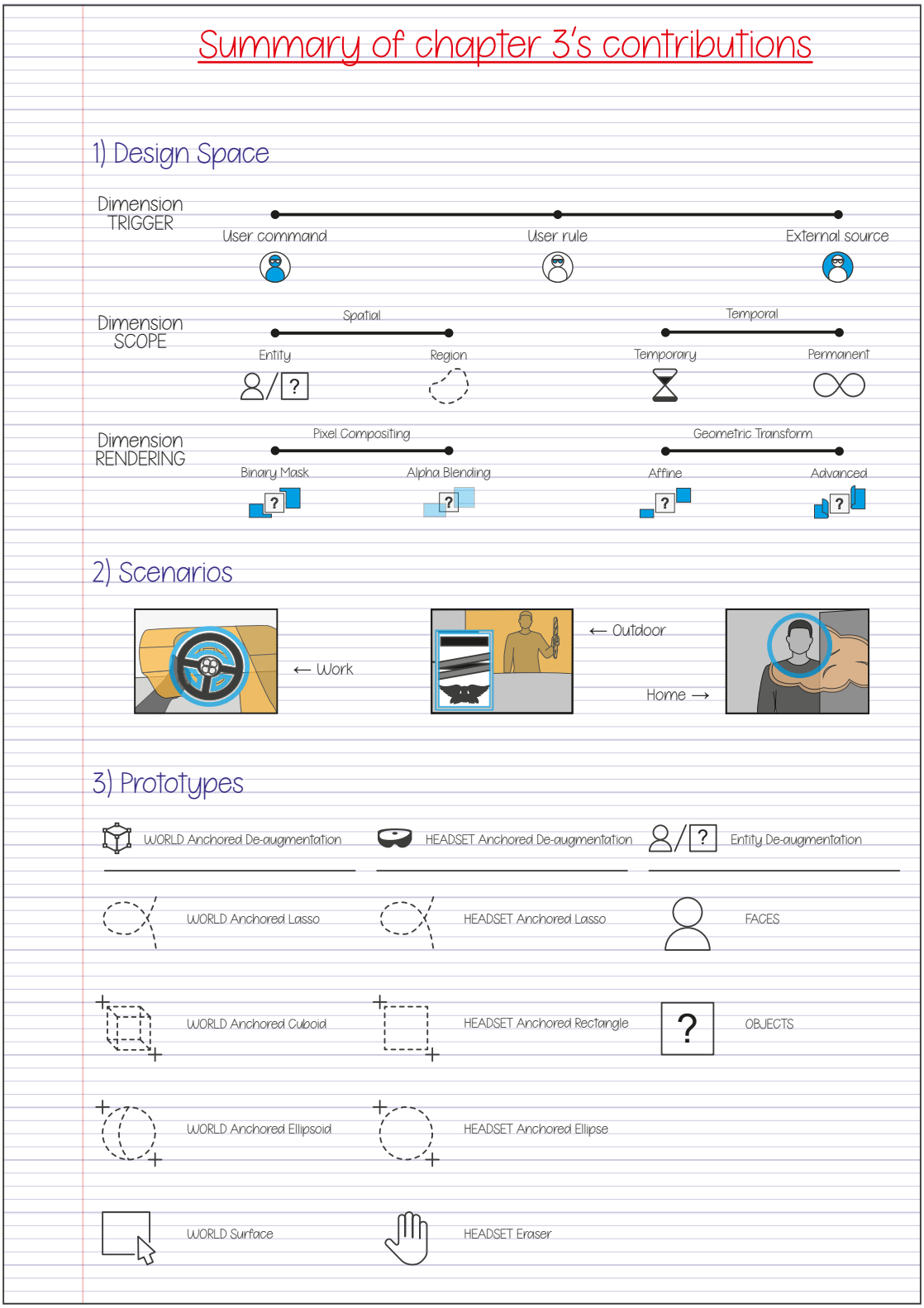


Fig. 3.9: Summary of chapter 3's contributions.

BIBLIOGRAPHY

[1] Adrien M & Claire B. Mirages & miracles, 2020.

[2] Ahlström, D., Hasan, K., and Irani, P. Are you comfortable doing that? acceptance studies of around-device gestures in and for public settings. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2014), MobileHCI '14, Association for Computing Machinery, pp. 193–202.

[3] Ahn, S., Heo, S., and Lee, G. Typing on a smartwatch for smart glasses. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (New York, New York, USA, 2017), ISS '17, Association for Computing Machinery, pp. 201–209.

[4] Anderson, F., Grossman, T., Wigdor, D., and Fitzmaurice, G. Supporting subtlety with deceptive devices and illusory interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 1489–1498.

[5] Appert, C., Chapuis, O., and Pietriga, E. High-precision magnification lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2010), CHI '10, Association for Computing Machinery, pp. 273–282.

[6] Appert, C., and Fekete, J.-D. Orthozoom scroller: 1d multi-scale navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2006), CHI '06, Association for Computing Machinery, pp. 21–30.

[7] Appert, C., Pietriga, E., Bartenlian, E., and González, R. M. Custom-made tangible interfaces with touchtokens. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (New York, New York, USA, 2018), AVI '18, Association for Computing Machinery, pp. 1–9.

[8] Arslan, C., Rekik, Y., and Grisoni, L. E-Pad: Large display pointing in a continuous interaction space around a mobile device. In *Proceedings of the Conference on Designing Interactive Systems* (New York, New York, USA, 2019), DIS '19, Association for Computing Machinery, pp. 1101–1108.

[9] Ashbrook, D., Baudisch, P., and White, S. Nenya: Subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2011), CHI '11, Association for Computing Machinery, pp. 2043–2046.

[10] Azmandian, M., Hancock, M., Benko, H., Ofek, E., and Wilson, A. D. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2016), CHI '16, Association for Computing Machinery, pp. 1968–1979.

[11] Azuma, R. T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6*, 4 (August 1997), 355–385.

[12] Bachynskyi, M., Palmas, G., Oulasvirta, A., Steimle, J., and Weinkauf, T. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 1817–1826.

[13] Bailly, G., Müller, J., Rohs, M., Wigdor, D., and Kratz, S. ShoeSense: A new perspective on gestural interaction and wearable applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2012), CHI '12, Association for Computing Machinery, pp. 1239–1248.

[14] Benko, H., Ofek, E., Zheng, F., and Wilson, A. D. Fovear: Combining an optically see-through near-eye display with projector-based spatial augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, New York, USA, 2015), UIST '15, Association for Computing Machinery, pp. 129–135.

[15] Billinghurst, M., Clark, A., and Lee, G. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction 8* (January 2015), 73–272.

[16] Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (New York, New York, USA, April 2004), CHI '04, Association for Computing Machinery, pp. 519–526.

[17] Botvinick, M., and Cohen, J. Rubber hands 'feel' touch that eyes see. *Nature 391*, 6669 (1998), 756.

[18] Brasier, E., Chapuis, O., Ferey, N., Vezien, J., and Appert, C. Arpads: Mid-air indirect input for augmented reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2020), ISMAR '20, pp. 332–343.

[19] Brasier, E., Pietriga, E., and Appert, C. Ar-enhanced widgets for smartphone-centric interaction. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction* (New York, New York, USA, September 2021), MobileHCI '21, Association for Computing Machinery.

[20] Brudy, F., Holz, C., Rädle, R., Wu, C.-J., Houben, S., Klokmose, C. N., and Marquardt, N. Cross-device taxonomy: Survey, opportunities and challenges of interactions spanning across multiple devices. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–28.

[21] Budhiraja, P., Sodhi, R., Jones, B., Karsch, K., Bailey, B., and Forsyth, D. Where's my drink? enabling peripheral real world interactions while using hmds. *arXiv preprint arXiv:1502.04744* (2015).

[22] Budhiraja, R., Lee, G. A., and Billinghurst, M. Interaction techniques for hmd-hhd hybrid ar systems. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2013), ISMAR '13, pp. 243–244.

[23] Budhiraja, R., Lee, G. A., and Billinghurst, M. Using a hhd with a hmd for mobile ar interaction. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2013), ISMAR '13, pp. 1–6.

[24] Butler, A., Izadi, S., and Hodges, S. Sidesight: Multi-"touch" interaction around small devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2008), UIST '08, Association for Computing Machinery, pp. 201–204.

[25] Butz, A., Beshers, C., and Feiner, S. Of vampire mirrors and privacy lamps: Privacy management in multi-user augmented environments. In *Proceedings*

of the 11th Annual ACM Symposium on User Interface Software and Technology (New York, New York, USA, 1998), UIST '98, Association for Computing Machinery, pp. 171–172.

[26] Buxton, W. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction* (NLD, 1990), INTERACT '90, North-Holland Publishing Co., pp. 449–456.

[27] Carrera, C. C., and Asensio, L. A. B. Augmented reality as a digital teaching environment to develop spatial thinking. *Cartography and Geographic Information Science 44*, 3 (2017), 259–270.

[28] Carrera, C. C., and Asensio, L. A. B. Landscape interpretation with augmented reality and maps to improve spatial orientation skill. *Journal of Geography in Higher Education 41*, 1 (2017), 119–133.

[29] Casiez, G., and Roussel, N. No more bricolage! methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2011), UIST '11, Association for Computing Machinery, pp. 603–614.

[30] Chang, C., Bang, K., Wetzstein, G., Lee, B., and Gao, L. Toward the next-generation vr/ar optics: a review of holographic near-eye displays from a human-centric perspective. *Optica 7*, 11 (November 2020), 1563–1578.

[31] Chapuis, O., and Dragicevic, P. Effects of motor scale, visual scale, and quantization on small target acquisition difficulty. *ACM Transactions on Computer-Human Interaction 18*, 3 (August 2011), 1–32.

[32] Chapuis, O., and Roussel, N. Copy-and-paste between overlapping windows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2007), CHI '07, Association for Computing Machinery, pp. 201–210.

[33] Chen, X. A., Marquardt, N., Tang, A., Boring, S., and Greenberg, S. Extending a mobile device's interaction space through body-centric interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2012), MobileHCI '12, Association for Computing Machinery, pp. 151–160.

[34] Chen, X. A., Schwarz, J., Harrison, C., Mankoff, J., and Hudson, S. E. Air+touch: Interweaving touch & in-air gestures. In *Proceedings*

of the 27th Annual ACM Symposium on User Interface Software and Technology (New York, New York, USA, 2014), UIST '14, Association for Computing Machinery, pp. 519–525.

[35] Chen, Y., Katsuragawa, K., and Lank, E. 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* (New York, New York, USA, 2020), CHI '20, Association for Computing Machinery, pp. 1–13.

[36] Colaço, A., Kirmani, A., Yang, H. S., Gong, N.-W., Schmandt, C., and Goyal, V. K. Mime: Compact, low power 3D gesture sensing for interaction with head mounted displays. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2013), UIST '13, Association for Computing Machinery, pp. 227–236.

[37] Courtoux, E., Appert, C., and Chapuis, O. Walltokens: Surface tangibles for vertical displays. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI '21, Association for Computing Machinery, pp. 1–13.

[38] Cyberpunk 2077, 2020.

[39] Dennō coil, 2007.

[40] Dobbelstein, D., Hock, P., and Rukzio, E. Belt: An unobtrusive touch input device for head-worn displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 2135–2138.

[41] Dobbelstein, D., Winkler, C., Haas, G., and Rukzio, E. Pocketthumb: A wearable dual-sided touch interface for cursor-based control of smart-eyewear. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (New York, New York, USA, June 2017), vol. 1 of *IMWUT '17*, Association for Computing Machinery, pp. 1–17.

[42] Dragicevic, P. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, October 2004), UIST '04, Association for Computing Machinery, pp. 193–196.

[43] Dudley, J., Benko, H., Wigdor, D., and Kristensson, P. O.

Performance envelopes of virtual keyboard text input strategies in virtual reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality* (2019), ISMAR '19, IEEE, pp. 289–300.

[44] Eiberger, A., Kristensson, P. O., Mayr, S., Kranz, M., and Grubert, J. Effects of depth layer switching between an optical see-through head-mounted display and a body-proximate display. In *Symposium on Spatial User Interaction* (New York, New York, USA, 2019), SUI '19, Association for Computing Machinery, pp. 1–9.

[45] Elmqvist, N., Henry, N., Riche, Y., and Fekete, J.-D. Melange: Space folding for multi-focus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2008), CHI '08, Association for Computing Machinery, pp. 1333–1342.

[46] Ens, B., Grossman, T., Anderson, F., Matejka, J., and Fitzmaurice, G. Candid interaction: Revealing hidden mobile and wearable computing activities. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, New York, USA, 2015), UIST '15, Association for Computing Machinery, pp. 467–476.

[47] Ens, B., Quigley, A., Yeo, H.-S., Irani, P., Piumsomboon, T., and Billinghurst, M. Counterpoint: Exploring mixed-scale gesture interaction for AR applications. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI EA '18, Association for Computing Machinery, pp. 1–6.

[48] Ens, B. M., Finnegan, R., and Irani, P. P. The personal cockpit: A spatial interface for effective task switching on head-worn displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2014), CHI '14, Association for Computing Machinery, pp. 3171–3180.

[49] Evers, C., Kniewel, R., Geihs, K., and Schmidt, L. Achieving user participation for adaptive applications. In *Ubiquitous Computing and Ambient Intelligence* (Berlin, Heidelberg, 2012), J. Bravo, D. López-de Ipiña, and F. Moya, Eds., UCAmI'12, Springer Berlin Heidelberg, pp. 200–207.

[50] Feiner, S., Macintyre, B., Höllerer, T., and Webster, A. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Digest of Papers. First International Symposium on Wearable Computers* (December 1997), vol. 1, pp. 74–81.

[51] Feiner, S., and Shamash, A. Hybrid user interfaces: Breeding virtually

bigger interfaces for physically smaller computers. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, November 1991), UIST '91, Association for Computing Machinery, pp. 9–17.

[52] Feuchtner, T., and Müller, J. Ownershift: Facilitating overhead interaction in virtual reality with an ownership-preserving hand space shift. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2018), UIST '18, Association for Computing Machinery, pp. 31–43.

[53] Forlines, C., Vogel, D., and Balakrishnan, R. Hybridpointing: Fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2006), UIST '06, Association for Computing Machinery, p. 211–220.

[54] Fruchard, B., Lecolinet, E., and Chapuis, O. Impact of semantic aids on command memorization for on-body interaction and directional gestures. In *Proceedings of the International Conference on Advanced Visual Interfaces* (New York, New York, USA, 2018), AVI '18, Association for Computing Machinery, pp. 1–9.

[55] Funk, M., Bächler, A., Bächler, L., Kosch, T., Heidenreich, T., and Schmidt, A. Working with augmented reality? a long-term analysis of in-situ instructions at the assembly workplace. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments* (New York, New York, USA, 2017), PETRA '17, Association for Computing Machinery, pp. 222–229.

[56] Gajos, K., and Weld, D. S. Supple: Automatically generating user interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (New York, New York, USA, January 2004), IUI '04, Association for Computing Machinery, pp. 93–100.

[57] Gervais, R., Frey, J., and Hachet, M. Pointing in spatial augmented reality from 2d pointing devices. In *Human-Computer Interaction* (September 2015), INTERACT '15, pp. 381–389.

[58] Google. Google glasses, 2012.

[59] Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., and Balakrishnan, R. Hover widgets: Using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems (New York, New York, USA, 2006), CHI '06, Association for Computing Machinery, pp. 861–870.

[60] Grubert, J., Heinisch, M., Quigley, A., and Schmalstieg, D. Multifi: Multi fidelity interaction with displays on and around the body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 3933–3942.

[61] Grubert, J., and Kranz, M. Headphones: Ad hoc mobile multi-display environments through head tracking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 3966–3971.

[62] Grubert, J., Kranz, M., and Quigley, A. Challenges in mobile multi-device ecosystems. *mUX: The Journal of Mobile User Experience 5*, 1 (2016), 1–22.

[63] Grubert, J., Ofek, E., Pahud, M., Kranz, M., and Schmalstieg, D. Glasshands: Interaction around unmodified mobile devices using sunglasses. In *Proceedings of the ACM International Conference on Interactive Surfaces and Spaces* (New York, New York, USA, 2016), ISS '16, Association for Computing Machinery, pp. 215–224.

[64] Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., and Kristensson, P. O. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018), VR '18, pp. 159–166.

[65] Gruenefeld, U., Stratmann, T. C., Jung, J., Lee, H., Choi, J., Nanda, A., and Heuten, W. Guiding smombies: Augmenting peripheral vision with low-cost glasses to shift the attention of smartphone users. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (October 2018), ISMAR-Adjunct '18, pp. 127–131.

[66] Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior 19*, 4 (1987), 486–517.

[67] Gustafson, S., Bierwirth, D., and Baudisch, P. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2010), UIST '10, Association for Computing Machinery, pp. 3–12.

[68] Ha, T., Feiner, S., and Woo, W. Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *IEEE International Symposium on Mixed and Augmented Reality* (September 2014), ISMAR '14, IEEE, pp. 219–228.

[69] Han, J., Ahn, S., Park, K., and Lee, G. Designing touch gestures using the space around the smartwatch as continuous input space. In *Proceedings of the ACM International Conference on Interactive Surfaces and Spaces* (New York, New York, USA, 2017), ISS '17, Association for Computing Machinery, pp. 210–219.

[70] Harrison, C., and Hudson, S. E. Abracadabra: Wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2009), UIST '09, Association for Computing Machinery, pp. 121–124.

[71] Harrison, C., Tan, D., and Morris, D. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2010), CHI '10, Association for Computing Machinery, pp. 453–462.

[72] Hartmann, J., Holz, C., Ofek, E., and Wilson, A. D. Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI'19, Association for Computing Machinery, pp. 1–12.

[73] Hasan, K., Ahlström, D., Kim, J., and Irani, P. Airpanes: Two-handed around-device interaction for pane switching on smartphones. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 679–691.

[74] Hettiarachchi, A., and Wigdor, D. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2016), CHI '16, Association for Computing Machinery, pp. 1957–1967.

[75] Hilliges, O., Kim, D., Izadi, S., Weiss, M., and Wilson, A. Holodesk: Direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2012), CHI '12, Association for Computing Machinery, pp. 2421–2430.

[76] Hincapié-Ramos, J. D., Guo, X., Moghadasian, P., and Irani, P. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2014), CHI '14, Association for Computing Machinery, pp. 1063–1072.

[77] Hincapié-Ramos, J. D., and Irani, P. Crashalert: Enhancing peripheral alertness for eyes-busy mobile interaction while walking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2013), CHI '13, Association for Computing Machinery, pp. 3385–3388.

[78] Hothorn, T., Hornik, K., van de Wiel, M. A., and Zeileis, A. Implementing a class of permutation tests: The coin package. *Journal of Statistical Software 28*, 8 (2008), 1–23.

[79] Hwang, S., Ahn, M., and Wohn, K.-y. Maggetz: Customizable passive tangible controllers on and around conventional mobile devices. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2013), UIST '13, Association for Computing Machinery, pp. 411–416.

[80] Iron man, 2008.

[81] Ishak, E. W., and Feiner, S. K. Interacting with hidden content using content-aware free-space transparency. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, October 2004), UIST '04, Association for Computing Machinery, pp. 189–192.

[82] ISO. 9241-9 ergonomic requirements for office work with visual display terminals (vdts)-part 9: Requirements for non-keyboard input devices. *International Organization for Standardization* (February 2000).

[83] Jacob, R. J. K. 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* (New York, New York, USA, March 1990), CHI '90, Association for Computing Machinery, pp. 11–18.

[84] Jacob, R. J. K. Eye tracking in advanced interface design. In *Virtual Environments and Advanced Interface Design*, W. Barfield and T. A. Furness, III, Eds. Oxford University Press, New York, New York, USA, 1995, pp. 258–288.

[85] Jones, B., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G. Around device interaction for multiscale navigation. In *Proceedings of the 14th*

International Conference on Human-Computer Interaction with Mobile Devices and Services (New York, New York, USA, 2012), MobileHCI '12, Association for Computing Machinery, pp. 83–92.

[86] Jota, R., Nacenta, M. A., Jorge, J. A., Carpendale, S., and Greenberg, S. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface 2010* (CAN, 2010), GI '10, Canadian Information Processing Society, pp. 269–276.

[87] Juan, M., Alcaniz, M., Monserrat, C., Botella, C., Banos, R., and Guerrero, B. Using augmented reality to treat phobias. *IEEE Computer Graphics and Applications 25*, 6 (December 2005), 31–37.

[88] Jung, J., Lee, H., Choi, J., Nanda, A., Gruenefeld, U., Stratmann, T., and Heuten, W. Ensuring safety in augmented reality from trade-off between immersion and situation awareness. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (October 2018), ISMAR '18, pp. 70–79.

[89] Kanamori, K., Sakata, N., Tominaga, T., Hijikata, Y., Harada, K., and Kiyokawa, K. Obstacle avoidance method in real space for virtual reality immersion. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (October 2018), ISMAR '18, pp. 80–89.

[90] Kanamori, K., Sakata, N., tominaga, T., Hijikata, Y., Harada, K., and Kiyokawa, K. Walking assist method for vr zombie. In *2019 12th Asia Pacific Workshop on Mixed and Augmented Reality (APMAR)* (March 2019), APMAR '19, pp. 15–21.

[91] Kang, H., Lee, G., and Han, J. Obstacle detection and alert system for smartphone ar users. In *25th ACM Symposium on Virtual Reality Software and Technology* (New York, New York, USA, 2019), VRST '19, Association for Computing Machinery, pp. 1–11.

[92] Kay, M., and Wobbrock, J. O. *ARTool: Aligned Rank Transform for Nonparametric Factorial ANOVAs*, 2020. R package version 0.10.7.

[93] Ketabdar, H., Roshandel, M., and Yüksel, K. A. Towards using embedded magnetic field sensor for around mobile device 3d interaction. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2010), MobileHCI '10, Association for Computing Machinery, pp. 153–156.

[94] Khalfa, S., Bruneau, N., Rogé, B., Georgieff, N., Veuillet, E., Adrien, J.-L., Barthélémy, C., and Collet, L. Increased perception of loudness in autism. *Hearing Research 198*, 1 (January 2005), 87–92.

[95] Khamis, M., Oechsner, C., Alt, F., and Bulling, A. VRpursuits: Interaction in virtual reality using smooth pursuit eye movements. In *Proceedings of the International Conference on Advanced Visual Interfaces* (New York, New York, USA, 2018), AVI '18, Association for Computing Machinery, pp. 1–8.

[96] Kim, J., He, J., Lyons, K., and Starner, T. The gesture watch: A wireless contact-free gesture based wrist interface. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers* (Washington, District of Columbia, USA, 2007), ISWC '07, IEEE, pp. 1–8.

[97] Knierim, P., Woundefinedniak, P. W., Abdelrahman, Y., and Schmidt, A. Exploring the potential of augmented reality in domestic environments. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2019), MobileHCI '19, Association for Computing Machinery, pp. 1–12.

[98] Koelle, M., El Ali, A., Cobus, V., Heuten, W., and Boll, S. C. All about acceptability? identifying factors for the adoption of data glasses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 295–300.

[99] Koelle, M., Kranz, M., and Möller, A. Don't look at me that way! understanding user attitudes towards data glasses usage. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2015), MobileHCI '15, Association for Computing Machinery, pp. 362–372.

[100] Kowalski, M., Nasarzewski, Z., Galinski, G., and Garbat, P. Holoface: Augmenting human-to-human interactions on hololens. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), WACV '18, pp. 141–149.

[101] Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., and Billinghurst, M. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1–14.

[102] Lalanne, D., Nigay, L., Palanque, p., Robinson, P., Vanderdonckt, J., and Ladry, J.-F. Fusion engines for multimodal input: A survey. In *Proceedings of the 2009 International Conference on Multimodal Interfaces* (New York, New York, USA, 2009), ICMI-MLMI '09, Association for Computing Machinery, pp. 153–160.

[103] Lan, S., Zhang, X., Taghinejad, M., Rodrigues, S., Lee, K.-T., Liu, Z., and Cai, W. Metasurfaces for near-eye augmented reality. *ACS Photonics 6*, 4 (April 2019), 864–870.

[104] Langlotz, T., Sutton, J., Zollmann, S., Itoh, Y., and Regenbrecht, H. Chromaglasses: Computational glasses for compensating colour blindness. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1–12.

[105] Laput, G., Xiao, R., Chen, X. A., Hudson, S. E., and Harrison, C. Skin buttons: Cheap, small, low-powered and clickable fixed-icon laser projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2014), UIST '14, Association for Computing Machinery, pp. 389–394.

[106] Lawrence, M. A. *ez: Easy Analysis and Visualization of Factorial Experiments*, 2016. R package version 4.4-0.

[107] Lee, D., Lee, Y., Shin, Y., and Oakley, I. Designing socially acceptable hand-to-face input. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2018), UIST '18, Association for Computing Machinery, pp. 711–723.

[108] Lee, G.-Y., Hong, J.-Y., Hwang, S., Moon, S., Kang, H., Jeon, S., Kim, H., Jeong, J.-H., and Lee, B. Metasurface eyepiece for augmented reality. *Nature Communications 9*, 1 (2018), 1–10.

[109] Lee, H., Kim, D., and Woo, W. Graphical menus using a mobile phone for wearable ar systems. In *2011 International Symposium on Ubiquitous Virtual Reality* (July 2011), ISUVR '11, pp. 55–58.

[110] Li, Z., Annett, M., Hinckley, K., Singh, K., and Wigdor, D. Holodoc: Enabling mixed reality workspaces that harness physical and digital content. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–14.

[111] Lindlbauer, D., Mueller, J., and Alexa, M. Changing the appearance of real-world objects by modifying their surroundings. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 3954–3965.

[112] Lindlbauer, D., and Wilson, A. D. Remixed reality: Manipulating space and time in augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1—13.

[113] Liu, M., Nancel, M., and Vogel, D. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, New York, USA, 2015), UIST '15, Association for Computing Machinery, pp. 63–71.

[114] Lobo, M.-J., Pietriga, E., and Appert, C. An evaluation of interactive map comparison techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 3573–3582.

[115] Long, B., Seah, S. A., Carter, T., and Subramanian, S. Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Transactions on Graphics 33*, 6 (November 2014), 1–10.

[116] Maggioni, E., Cobden, R., and Obrist, M. Owidgets: A toolkit to enable smell-based experience design. *International Journal of Human-Computer Studies 130* (June 2019), 248–260.

[117] Malacria, S., Scarr, J., Cockburn, A., Gutwin, C., and Grossman, T. Skillometers: Reflective widgets that motivate and help users to improve performance. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2013), UIST '13, Association for Computing Machinery, pp. 321–330.

[118] Marquardt, N., Brudy, F., Liu, C., Bengler, B., and Holz, C. Surfaceconstellations: A modular hardware platform for ad-hoc reconfigurable cross-device workspaces. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1–14.

[119] Marquardt, N., Jota, R., Greenberg, S., and Jorge, J. A. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In *Human-Computer Interaction – INTERACT 2011*

(Berlin, Heidelberg, 2011), P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds., INTERACT '11, Springer Berlin Heidelberg, pp. 461–476.

[120] Marwecki, S., Wilson, A. D., Ofek, E., Gonzalez Franco, M., and Holz, C. Mise-unseen: Using eye tracking to hide virtual reality scene changes in plain sight. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2019), UIST '19, Association for Computing Machinery, pp. 777–789.

[121] Matsuda, K. Hyper-reality. Short film, 2016.

[122] May-raz, E., and Lazo, D. Sight. Short film, 2012.

[123] McCallum, D. C., and Irani, P. Arc-pad: Absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2009), UIST '09, Association for Computing Machinery, pp. 153–156.

[124] McGill, M., Boland, D., Murray-Smith, R., and Brewster, S. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI '15, Association for Computing Machinery, pp. 2143–2152.

[125] McGuffin, M. Augmented reality knowledge work : Towards a research agenda. In *Extended Abstracts of ACM CHI Conference on Human Factors in Computing Systems (CHI 2019)* (2019), CHI EA '19.

[126] Menzner, T., Gesslein, T., Otte, A., and Grubert, J. Above surface interaction for multiscale navigation in mobile virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2020), VR '20, pp. 372–381.

[127] Microsoft. Microsoft hololens, 2017.

[128] Milgram, P., and Kishino, F. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems vol. E77-D, no. 12* (December 1994), 1321–1329.

[129] Miller, M. R., Jun, H., Herrera, F., Yu Villa, J., Welch, G., and Bailenson, J. N. Social interaction in augmented reality. *PLOS ONE 14*, 5 (May 2019), 1–26.

[130] Millette, A., and McGuffin, M. J. Dualcad: Integrating augmented reality with a desktop gui and smartphone interaction. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)* (2016), ISMAR-Adjunct '16, pp. 21–26.

[131] Min Htike, H., H. Margrain, T., Lai, Y.-K., and Eslambolchilar, P. Augmented reality glasses as an orientation and mobility aid for people with low vision: A feasibility study of experiences and requirements. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI '21, Association for Computing Machinery, pp. 1–15.

[132] Mine, M. R., Brooks, Jr., F. P., and Sequin, C. H. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, New York, USA, 1997), SIGGRAPH '97, Association for Computing Machinery & Addison-Wesley, pp. 19–26.

[133] Minority report, 2002.

[134] Mistry, P., and Maes, P. Sixthsense: A wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches* (New York, New York, USA, 2009), SIGGRAPH ASIA '09, Association for Computing Machinery, p. 1.

[135] Mohr, P., Tatzgern, M., Langlotz, T., Lang, A., Schmalstieg, D., and Kalkofen, D. Trackcap: Enabling smartphones for 3d interaction on mobile head-mounted displays. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–11.

[136] Mojo. Mojo lens.

[137] Montano Murillo, R. A., Subramanian, S., and Martinez Plasencia, D. Erg-O: Ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2017), UIST '17, Association for Computing Machinery, pp. 759–771.

[138] Morales González, R., Appert, C., Bailly, G., and Pietriga, E. Touchtokens: Guiding touch patterns with passive tokens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2016), CHI '16, Association for Computing Machinery, pp. 4189–4202.

[139] Morris, M. R., Danielescu, A., Drucker, S., Fisher, D., Lee,

B., Wobbrock, J. O., et al. Reducing legacy bias in gesture elicitation studies. *Interactions 21*, 3 (2014), 40–45.

[140] Müller, F., Dezfuli, N., Mühlhäuser, M., Schmitz, M., and Khalilbeigi, M. Palm-based interaction with head-mounted displays. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (New York, New York, USA, 2015), MobileHCI '15, Association for Computing Machinery, pp. 963–965.

[141] Müller, F., McManus, J., Günther, S., Schmitz, M., Mühlhäuser, M., and Funk, M. Mind the tap: Assessing foot-taps for interacting with head-mounted displays. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–13.

[142] Nancel, M., Pietriga, E., Chapuis, O., and Beaudouin-Lafon, M. Mid-air pointing on ultra-walls. *ACM Transactions on Computer-Human Interaction 22*, 5 (August 2015), 1–62.

[143] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2011), CHI '11, Association for Computing Machinery, pp. 177–186.

[144] Neuwirth, E. *RColorBrewer: ColorBrewer Palettes*, 2014. R package version 1.1-2.

[145] Niantic. Pokémon go, 2016.

[146] Nielsen Norman Group. Mobile user experience: Limitations and strengths. Website, 2015.

[147] Nigay, L., and Coutaz, J. A design space for multimodal systems: Concurrent processing and data fusion. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (New York, New York, USA, May 1993), CHI '93, Association for Computing Machinery, pp. 172–178.

[148] Normand, E., and McGuffin, M. J. Enlarging a smartphone with ar to create a handheld vesad (virtually extended screen-aligned display). In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2018), ISMAR '18, pp. 123–133.

[149] Ostberg, A., and Matic, N. Hover cursor: Improving touchscreen acquisition of small targets with hover-enabled pre-selection. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, New York, USA, 2015), CHI EA '15, Association for Computing Machinery, pp. 1723–1728.

[150] Parviz, B. A. Augmented reality in a contact lens. *IEEE Spectrum* (September 2009).

[151] Peddie, J. *Augmented reality: Where we will all live*. Springer, 2017.

[152] Pedersen, I., Gale, N., Mirza-Babaei, P., and Reid, S. More than meets the eye: The benefits of augmented reality and holographic displays for digital cultural heritage. *Journal on Computing and Cultural Heritage 10*, 2 (March 2017), 1–15.

[153] Perry, T. S. Augmented reality in a contact lens: It's the real deal. *IEEE Spectrum* (January 2020).

[154] Pezent, E., O'Malley, M. K., Israr, A., Samad, M., Robinson, S., Agarwal, P., Benko, H., and Colonnese, N. Explorations of wrist haptic feedback for ar/vr interactions with tasbi. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2020), CHI EA '20, Association for Computing Machinery, pp. 1–4.

[155] Pfeuffer, K., Alexander, J., Chong, M. K., Zhang, Y., and Gellersen, H. Gaze-Shifting: Direct-Indirect input with pen and touch modulated by gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, New York, USA, 2015), UIST '15, Association for Computing Machinery, pp. 373–383.

[156] Pfeuffer, K., Mayer, B., Mardanbegi, D., and Gellersen, H. Gaze + Pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (New York, New York, USA, 2017), SUI '17, Association for Computing Machinery, pp. 99–108.

[157] Piekarski, W., and Thomas, B. Arquake: The outdoor augmented reality gaming system. *Communications of the ACM 45*, 1 (January 2002), 36–38.

[158] Pierce, J. S., Forsberg, A. S., Conway, M. J., Hong, S., Zeleznik, R. C., and Mine, M. R. Image plane interaction techniques in 3D immersive environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (New York, New York, USA, 1997), I3D '97, Association for Computing

Machinery, pp. 39–43.

[159] Pietroszek, K., Tahai, L., Wallace, J. R., and Lank, E. Watchcasting: Freehand 3d interaction with off-the-shelf smartwatch. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)* (2017), 3DUI '17, pp. 172–175.

[160] Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. Jellylens: Content-aware adaptive lenses. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2012), UIST '12, Association for Computing Machinery, pp. 261–270.

[161] Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. Drilling into complex 3d models with gimlenses. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology* (New York, New York, USA, 2013), VRST '13, Association for Computing Machinery, pp. 223–230.

[162] Piumsomboon, T., Altimira, D., Kim, H., Clark, A., Lee, G., and Billinghurst, M. Grasp-Shell vs Gesture-Speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *International Symposium on Mixed and Augmented Reality* (September 2014), ISMAR '14, IEEE, pp. 73–82.

[163] Piumsomboon, T., Lee, G., Lindeman, R. W., and Billinghurst, M. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *IEEE Symposium on 3D User Interfaces* (March 2017), 3DUI '17, IEEE, pp. 36–39.

[164] Pohl, H., and Murray-Smith, R. Focused and casual interactions: Allowing users to vary their level of engagement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2013), CHI '13, Association for Computing Machinery, pp. 2223–2232.

[165] Poulton, E. C. *Tracking skill and manual control*. Academic press, December 1974.

[166] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. The Go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 1996), UIST '96, Association for Computing Machinery, pp. 79–80.

[167] R Core Team. R: A language and environment for statistical computing, 2018.

[168] Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H., and

Rogers, Y. Huddlelamp: Spatially-aware mobile displays for ad-hoc around-the-table collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces* (New York, New York, USA, 2014), ITS '14, Association for Computing Machinery, pp. 45–54.

[169] Radu, I. Why should my students use ar? a comparative review of the educational impacts of augmented-reality. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2012), pp. 313–314.

[170] Ranasinghe, N., Jain, P., Karwita, S., Tolley, D., and Do, E. Y.-L. Ambiotherm: Enhancing sense of presence in virtual reality by simulating real-world environmental conditions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 1731–1742.

[171] Rateau, H., Rekik, Y., Lank, E., and Grisoni, L. Ether-toolbars: Evaluating off-screen toolbars for mobile interaction. In *23rd International Conference on Intelligent User Interfaces* (New York, New York, USA, 2018), IUI '18, Association for Computing Machinery, pp. 487–495.

[172] Razzaque, S., Kohn, Z., and Whitton, M. C. Redirected walking. In *Eurographics 2001 - Short Presentations* (August 2001), Eurographics Association.

[173] Reipschläger, P., and Dachselt, R. Designar: Immersive 3d-modeling combining augmented reality with interactive displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (New York, New York, USA, 2019), ISS '19, Association for Computing Machinery, pp. 29–41.

[174] Rixen, J. O., Hirzle, T., Colley, M., Etzel, Y., Rukzio, E., and Gugenheimer, J. Exploring augmented visual alterations in interpersonal communication. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI '21, Association for Computing Machinery, pp. 1–11.

[175] Roesner, F., Kohno, T., and Molnar, D. Security and privacy for augmented reality systems. *Communications of the ACM 57*, 4 (April 2014), 88–96.

[176] Rogers, Y. Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. In *UbiComp 2006: Ubiquitous Computing* (Berlin, Heidelberg, 2006), P. Dourish and A. Friday, Eds., Springer Berlin Heidelberg, pp. 404–421.

[177] Saidi, H., Serrano, M., Irani, P., Hurter, C., and Dubois, E. On-body tangible interaction: Using the body to support tangible manipulations

for immersive environments. In *Proceedings of the 17th IFIP TC 13 International Conference on Human-computer Interaction* (Cham, 2019), D. Lamas, F. Loizides, L. Nacke, H. Petrie, M. Winckler, and P. Zaphiris, Eds., INTERACT'19, Springer, pp. 471–492.

[178] Schlosser, P. Head-worn displays for emergency medical services. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI EA '21, Association for Computing Machinery, pp. 1–5.

[179] Schlosser, P., Matthews, B., Salisbury, I., Sanderson, P., and Hayes, S. Head-worn displays for emergency medical services staff: Properties of prehospital work, use cases, and design considerations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI '21, Association for Computing Machinery, pp. 1–14.

[180] Serrano, M., Ens, B., Yang, X.-D., and Irani, P. Gluey: Developing a head-worn display interface to unify the interaction experience in distributed display environments. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2015), MobileHCI '15, Association for Computing Machinery, pp. 161–171.

[181] Serrano, M., Ens, B. M., and Irani, P. P. Exploring the use of hand-to-face input for interacting with head-worn displays. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (New York, New York, USA, 2014), CHI '14, Association for Computing Machinery, pp. 3181–3190.

[182] Serrano, M., Hildebrandt, D., Subramanian, S., and Irani, P. Identifying suitable projection parameters and display configurations for mobile true-3d displays. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services* (New York, New York, USA, 2014), MobileHCI '14, Association for Computing Machinery, pp. 135–143.

[183] Serrano, M., Nigay, L., Lawson, J.-Y. L., Ramsay, A., Murray-Smith, R., and Denef, S. The openinterface framework: A tool for multimodal interaction. In *Extended Abstracts of the 2008 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2008), CHI EA '08, Association for Computing Machinery, pp. 3501–3506.

[184] Sicat, R., Li, J., Choi, J., Cordeil, M., Jeong, W.-K., Bach, B., and Pfister, H. DXR: A toolkit for building immersive data visualizations.

IEEE Transactions on Visualization and Computer Graphics 25, 1 (January 2019), 715–725.

[185] Siddhpuria, S., Katsuragawa, K., Wallace, J. R., and Lank, E. Exploring at-your-side gestural interaction for ubiquitous environments. In *Proceedings of the Conference on Designing Interactive Systems* (New York, New York, USA, 2017), DIS '17, Association for Computing Machinery, pp. 1111–1122.

[186] Siddhpuria, S., Malacria, S., Nancel, M., and Lank, E. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1–11.

[187] Silva-Néto, R., and Soares, A. Osmophobia and odour-triggered headaches – review of the literature and main research centres. *European Neurological Review 12* (March 2017), 24–27.

[188] Slater, M., Gonzalez-Liencres, C., Haggard, P., Vinkers, C., Gregory-Clarke, R., Jelley, S., Watson, Z., Breen, G., Schwarz, R., Steptoe, W., Szostak, D., Halan, S., Fox, D., and Silver, J. The ethics of realism in virtual and augmented reality. *Frontiers in Virtual Reality 1* (2020), 1.

[189] Sodnik, J., Tomazic, S., Grasset, R., Duenser, A., and Billinghurst, M. Spatial sound localization in an augmented reality environment. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments* (New York, New York, USA, 2006), OZCHI '06, Association for Computing Machinery, p. 111–118.

[190] Spelmezan, D., Appert, C., Chapuis, O., and Pietriga, E. Controlling widgets with one power-up button. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2013), UIST '13, Association for Computing Machinery, pp. 71–74.

[191] Spelmezan, D., Appert, C., Chapuis, O., and Pietriga, E. Side pressure for bidirectional navigation on small devices. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, New York, USA, 2013), MobileHCI '13, Association for Computing Machinery, pp. 11–20.

[192] Starner, T., Auxier, J., Ashbrook, D., and Gandy, M. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Proceedings of the 4th*

IEEE International Symposium on Wearable Computers (Washington, District of Columbia, USA, 2000), ISWC '00, IEEE, pp. 87–94.

[193] Stellmach, S., and Dachselt, R. Look & Touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2012), CHI '12, Association for Computing Machinery, pp. 2981–2990.

[194] Stephenson, N. Le samouraï virtuel, 1992.

[195] Stuerzlinger, W., Chapuis, O., Phillips, D., and Roussel, N. User interface façades: Towards fully adaptable user interfaces. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2006), UIST '06, Association for Computing Machinery, p. 309–318.

[196] Surale, H. B., Gupta, A., Hancock, M., and Vogel, D. Tabletinvr: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2019), CHI '19, Association for Computing Machinery, pp. 1–13.

[197] Sutherland, I. E. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I* (New York, New York, USA, 1968), AFIPS '68 (Fall, part I), Association for Computing Machinery, pp. 757–764.

[198] Syiem, B. V., Kelly, R. M., Goncalves, J., Velloso, E., and Dingler, T. Impact of task on attentional tunneling in handheld augmented reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2021), CHI '21, Association for Computing Machinery, pp. 1–14.

[199] Tan, D. S., Meyers, B., and Czerwinski, M. Wincuts: Manipulating arbitrary window regions for more effective use of screen space. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (New York, New York, USA, January 2004), CHI EA '04, Association for Computing Machinery, p. 1525–1528.

[200] Tecchia, F., Avveduto, G., Brondi, R., Carrozzino, M., Bergamasco, M., and Alem, L. I'm in vr! using your own hands in a fully immersive mr system. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (New York, New York, USA, 2014), VRST '14, Association for Computing Machinery, pp. 73–76.

[201] The expanse, 2015.

[202] Ungureanu, D., Bogo, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T. J., Tekin, B., Schönberger, J. L., Tekin, B., Olszta, P., and Pollefeys, M. Hololens 2 research mode as a tool for computer vision research. *arXiv:2008.11239* (2020).

[203] Varjo. Varjo xr-3, 2020.

[204] Vi, C. T., Ablart, D., Arthur, D., and Obrist, M. Gustatory interface: The challenges of 'how' to stimulate the sense of taste. In *Proceedings of the 2nd ACM SIGCHI International Workshop on Multisensory Approaches to Human-Food Interaction* (New York, New York, USA, 2017), MHFI '17, Association for Computing Machinery, pp. 29–33.

[205] Vo, D.-B., Lecolinet, E., and Guiard, Y. Belly gestures: Body centric gestures on the abdomen. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (New York, New York, USA, 2014), NordiCHI '14, Association for Computing Machinery, pp. 687–696.

[206] Vogel, D., and Baudisch, P. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2007), CHI '07, Association for Computing Machinery, pp. 657–666.

[207] Waldner, M., Steinberger, M., Grasset, R., and Schmalstieg, D. Importance-driven compositing window management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, May 2011), CHI '11, Association for Computing Machinery, pp. 959–968.

[208] Ware, C., and Mikaelian, H. H. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface* (New York, New York, USA, May 1987), CHI '87, Association for Computing Machinery, pp. 183–188.

[209] Weld, D. S., Anderson, C., Domingos, P., Etzioni, O., Gajos, K., Lau, T., and Wolfman, S. Automatically personalizing user interfaces. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (San Francisco, CA, USA, 2003), IJCAI'03, Morgan Kaufmann Publishers Inc., pp. 1613–1619.

[210] Wellner, P. Interacting with paper on the digitaldesk. *Communications of the ACM 36*, 7 (July 1993), 87–96.

[211] Wenig, D., Schöning, J., Olwal, A., Oben, M., and Malaka, R. Watchthru: Expanding smartwatch displays with mid-air visuals and wrist-worn augmented reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2017), CHI '17, Association for Computing Machinery, pp. 716–721.

[212] Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[213] Wolf, D., Dudley, J. J., and Kristensson, P. O. Performance envelopes of in-air direct and smartwatch indirect control for head-mounted augmented reality. In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018), VR '18, pp. 347–354.

[214] Yamashita, K., Kikuchi, T., Masai, K., Sugimoto, M., Thomas, B. H., and Sugiura, Y. CheekInput: Turning your cheek into an input surface by embedded optical sensors on a head-mounted display. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology* (New York, New York, USA, 2017), VRST '17, Association for Computing Machinery, pp. 1–8.

[215] Yang, K.-T., Wang, C.-H., and Chan, L. Sharespace: Facilitating shared use of the physical space by both vr head-mounted display and external users. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2018), UIST '18, Association for Computing Machinery, pp. 499–509.

[216] Yue, Y.-T., Yang, Y.-L., Ren, G., and Wang, W. Scenectrl: Mixed reality enhancement via efficient scene editing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA, 2017), UIST '17, Association for Computing Machinery, pp. 427–436.

[217] Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 1999), CHI '99, Association for Computing Machinery, pp. 246–253.

[218] Zhu, F., and Grossman, T. Bishare: Exploring bidirectional interactions between smartphones and head-mounted augmented reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (New York, New York, USA, 2020), CHI '20, Association for Computing Machinery, pp. 1–14.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Usages de la Réalité Augmentée pour la Vie Quotidienne

Mots clés: Interaction Homme-Machine, Réalité Augmentée

Résumé:
Ce manuscrit présente les recherches que j'ai effectuées dans le cadre de ma thèse de doctorat autour des usages de la Réalité Augmentée (RA) dans la vie de tous les jours. J'y présente trois questions de recherche :

QR_1 Comment reconsidérer nos interactions avec la Réalité Augmentée portative pour que celles-ci soient appropriées pour des interactions prolongées ?

QR_2 Peut-on énoncer des recommandations sur l'augmentation des appareils mobiles avec la Réalité Augmentée portative ?

QR_3 Comment un utilisateur peut-il garder le contrôle sur le contenu qu'il perçoit en Réalité Augmentée ?

Le manuscrit est organisé en trois parties répondant respectivement à chacune de ces trois questions de recherche.

Pour répondre à notre première question (QR_1), nous présentons le projet ARPads. Un ARPad n'a pas de réalité physique, mais représente un plan d'interaction flottant dans l'air dans lequel les mouvements de l'utilisateur sont traduits par des mouvements de curseur dans une fenêtre de RA affichée dans l'environnement. Ce contrôle indirect permet à l'utilisateur d'adopter une posture plus confortable pour manipuler du contenu affiché en RA par opposition au contrôle direct qui force l'utilisateur à maintenir ses bras en l'air en direction du contenu de RA. Après avoir défini les critères de conceptions de ces ARPads en prenant en compte leur position et leur orientation relatives à l'utilisateur, nous avons implémenté et évalué certaines conceptions prometteuses. Nos résultats montrent que le contrôle indirect peut atteindre des performances similaires à celles du contrôle direct, tout en limitant la fatigue perçue par l'utilisateur et identifient les conceptions d'ARPads les plus efficaces.

Pour notre deuxième question (QR_2), nous nous concentrons sur la combinaison entre les appareils mobiles et la RA. Suite à un atelier de conception mené avec des utilisateurs d'appareils mobiles, nous avons défini un espace de conception d'applications centrées sur le smartphone pouvant bénéficier d'une association avec la RA. En particulier, nous avons identifié deux dimensions principales : la fonction et la position du contenu holographique par rapport au contenu de premier intérêt affiché sur l'appareil mobile. Ce premier travail montre une grande richesse en termes de créativité pour proposer des améliorations possibles pour les appareils mobiles. Cependant, le manque de données quantifiables sur leur performance effective nous empêche encore de poser un avis objectif sur les réels avantages d'une telle association. Après avoir prototypé quelques exemples d'application pour démontrer leur faisabilité, nous nous sommes concentrés sur l'évaluation d'un cas spécifique : les composants d'interface (widgets) traditionnels des smartphones.

Enfin, pour notre troisième question (QR_3), nous nous projetons dans un futur dans lequel le champ de vision des utilisateurs contient fréquemment du contenu virtuel, et proposons une solution concrète pour que celui-ci n'interfère pas avec leur perception du monde réel. Pour cela, nous introduisons le concept de de-augmentation de la RA. Il s'agit de permettre aux utilisateurs de retirer volontairement certains éléments holographiques de leur champ de vision. Nous commençons par définir une taxonomie des opérations de dé-augmentation qui s'articule autour de trois dimensions : l'objet de la dé-augmentation, son déclencheur, et son rendu graphique. Nous illustrons ensuite ce concept au travers de trois scénarios démontrant l'utilité de telles opérations dans quelques situations particulières. Finalement, nous proposons une implémentation pratique de ce concept avec un prototype opérationnel, dont nous détaillons les différentes interactions qui permettent à l'utilisateur de définir une zone dé-augmentée. Ce dernier projet, plus théorique, ouvre des questionnements et des perspectives sur l'orientation des futurs systèmes de RA.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Using Augmented Reality in Everyday Life

Keywords: Human-Computer Interaction, Augmented Reality

Summary:
This manuscript describes the research topics I explored during my Ph.D. regarding the uses of Augmented Reality (AR) in everyday life. I tackle the three following research questions:

RQ_1 How to reconsider interaction with wearable AR to better support long-lasting tasks?

RQ_2 Can we provide guidelines on how a wearable AR device can enhance a handheld device?

RQ_3 How can users be in control of what Augmented Reality tells them?

The manuscript is divided into three parts respectively addressing each of these three research questions.

To answer our first research question ((RQ_1)), we present the ARPads project. An ARPad has no tangibility but represents a floating interaction plane on which users can move their hands to control a cursor displayed in an AR window. Such indirect input allows users to keep a comfortable posture while interacting with AR, as opposed to direct input that forces users to keep their arms in an upward position towards the content displayed in AR in front of them. After exploring a design space of these ARPads regarding their position and orientation relative to the user, we implement and empirically evaluate some promising combinations. Our results show that indirect input can achieve the same performance as direct input while limiting users' fatigue. From these results, we derive guidelines for future implementations.

Regarding our second research question (RQ_2), we focus on the association of wearable AR and mobile devices. We adopt a user-centered approach with a workshop organized with users of mobile devices. Based on the feedback derived from this workshop, we define a design space of smartphone-centric applications that could benefit from this association. In particular, we identify two main dimensions: the function and the location of the AR content relative to the main content displayed on the mobile device's screen. This first contribution highlights the creative ways to enhance mobile devices with AR. However, not much can be asserted regarding such enhancements without actual measures about their performance. After prototyping some use cases to show their feasibility, we evaluate some UI component distributions between a phone's screen and AR.

Finally, we answer our last question (RQ_3) by anticipating a future where users' field of view is frequently augmented with AR content. We introduce the concept of AR de-augmentation as a means to prevent this AR content from interfering with users' perception of the real world. AR de-augmentation gives users agency over the AR content. We first define a taxonomy of such de-augmentation operations regarding three aspects: its scope, its trigger, and its rendering. Then, we illustrate the concept of AR de-augmentation with three scenarios to demonstrate its usefulness in some specific use cases. Finally, we implement a working prototype in which we detail some interactions that allow users to define a de-augmented area. This last project is more theoretical and projective, opening questions and perspectives on future research directions.