



HAL
open science

Awareness Model for Asymmetric Remote Collaboration in Mixed Reality

Morgan Le Chénéchal

► **To cite this version:**

Morgan Le Chénéchal. Awareness Model for Asymmetric Remote Collaboration in Mixed Reality. Human-Computer Interaction [cs.HC]. INSA Rennes, 2016. English. NNT: . tel-01393627

HAL Id: tel-01393627

<https://hal.science/tel-01393627>

Submitted on 7 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse

UNIVERSITE
BRETAGNE
LOIRE

THÈSE INSA de Rennes
sous le sceau de l'Université Bretagne Loire

pour obtenir le grade de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Informatique

présentée par

Morgan LE CHÉNÉCHAL

ÉCOLE DOCTORALE : MATISSE
LABORATOIRE : IRISA – UMR6074

Awareness Model for Asymmetric Remote Collaboration in Mixed Reality

Soutenance prévue le 21 juillet 2016

devant le jury composé de :

Sabine COQUILLART

Directeur de recherche à INRIA Grenoble / *Rapporteuse*

Guillaume MOREAU

Professeur à l'École Centrale de Nantes / *Rapporteur*

Torsten KUHLEN

Professeur à l'Université de RWTH Aachen / *Examinateur*

Laurence NIGAY

Professeur à l'Université de Grenoble / *Examinatrice*

Bruno ARNALDI

Professeur à l'INSA de Rennes / *Directeur*

Thierry DUVAL

Professeur à Télécom Bretagne / *Co-encadrant*

Valérie GOURANTON

Maître de conférences à l'INSA de Rennes / *Co-encadrante*

Jérôme ROYAN

Chef de laboratoire R&D à b<>com / *Co-encadrant*

Awareness Model for Asymmetric Remote Collaboration in Mixed Reality

Morgan Le Chénéchal



En partenariat avec



Contents

Introduction	1
1 Related Work	5
1.1 Synchronous Remote Collaboration	5
1.1.1 Asymmetry Dimensions	5
1.1.2 Fields of Remote Collaboration	8
1.2 Interaction Techniques	11
1.2.1 Classification	11
1.2.2 Interaction Metaphors	12
1.2.3 Interaction Scheme	12
1.2.4 Collaborative Interactions	13
1.3 Framework for Remote Collaboration	14
1.3.1 Shared World Synchronization	14
1.3.2 Multi-Scale	15
1.3.3 Collaborative and Concurrent Interaction	16
1.3.4 Summary	17
1.4 State of the art analysis	18
1.4.1 Mixed Reality	18
1.4.2 Remote Collaboration	18
1.4.3 Challenges & Contributions Overview	19
2 Awareness	21
2.1 State of the Art	21
2.1.1 Awareness classification	22
2.1.2 Awareness basics: VE and content	23
2.1.3 Multi-modal awareness	24
2.1.4 Awareness techniques for CVE	24
2.2 A new generic awareness model	26
2.2.1 Single user	26
2.2.2 Collaborative setting	28
2.2.3 Multi-users MR setting	30
2.3 Users Representation	33
2.4 User's Point of View	34
2.5 Preliminary user evaluation	36
2.5.1 Use case overview: Asymmetric VR remote guiding	36
2.5.2 Experimental apparatus	38
2.5.3 Results	41

2.5.4	Experimental design comments	45
2.5.5	Discussion	46
2.5.6	Awareness proposal: an additional asymmetric loop	46
2.6	Conclusion	48
3	Remote guiding	49
3.1	Introduction	49
3.2	State of the art	50
3.2.1	Navigation	51
3.2.2	Manipulation	52
3.3	Proposed approaches	52
3.4	Asymmetrical setting: HWD and WIM	54
3.4.1	Implementation	58
3.4.2	Navigation guiding	58
3.5	Symmetrical setting: Fully collocated	66
3.5.1	System design	69
3.5.2	Pilot user study	71
3.5.3	Navigation guiding	77
3.6	Conclusion	82
4	Co-manipulation	85
4.1	State of the Art	85
4.2	Generic Aggregation-based with Heterogeneous Devices	86
4.2.1	Model description	86
4.2.2	Visual feedbacks	89
4.2.3	Asymmetrical devices and PoVs	89
4.2.4	Improvement tracks	89
4.3	Multi-Scale Hybrid Split of DoFs	91
4.3.1	Motivations	91
4.3.2	Overview	94
4.3.3	Asymmetric collaborative scenario	94
4.3.4	Interaction Techniques	96
4.3.5	Implementation	100
4.3.6	Preliminary Tests	101
4.3.7	Conclusion & Perspectives	102
	Conclusion	103
	Contributions	104
	Perspectives	105
	List of Figures	109
	List of Tables	113
	List of Acronyms	115
	Appendix A The CollabManager Framework	117
	A.1 Architecture overview	118

A.2 Documentation: How to use CollabMgr for Unity3D?	119
Publications	127
Bibliography	129

Introduction

Remote collaboration is interesting in many cases because it provides ways to collaborate with geographically distant people without needing to travel, thus saving time and money. This kind of collaboration can be synchronous, with distant people collaborating in real-time, or asynchronous, which allows flexibility about the time the different collaborators act on their shared data and content (especially it can be interesting to handle time zones difference issues). In our work, we focus on synchronous remote collaboration with the idea to offer ways to collaborate as effectively as if collaborators were around the same table in the same room.

The development of high-end networks provides more and more bandwidth, while decreasing latency and improving Quality of Service (QoS). Moreover, nowadays, we assist to the emergence of innovative technologies and usages driven by the maturity of these technologies and their availability on the consumer market. Particularly, Mixed Reality [Milgram and Kishino, 1994] technologies are able to mix real and virtual content in order to provide immersive user experience while interacting in a more natural way, compared with more classical desktop setups, which can improve efficiency and usability of these applications. These technologies allow to imagine new kinds of systems to handle computer-mediated remote collaboration by providing new dimensions to this field. MR adds natural capabilities to remote collaboration, in terms of interaction as well as perception. In many cases it can improve the collaborative process by providing enhanced interaction and perception possibilities taking advantages of the human skills in performing 3D tasks in his/her everyday life in the real world.

In this PhD thesis, we focus on **synchronous asymmetric remote collaboration applied to maintenance**. As an example, if an operator who has to perform a physical task in the real world needs to be helped by a remote expert, we can imagine a system based on MR technologies in order to virtually collocate the remote expert in the operator's workspace (i.e to virtually bring the remote expert within the geographically distant operator's workspace). Then, the expert should be able to help the operator in performing his/her task in a natural way using easy-to-understand helping cues.

Our context opens different fields of investigation:

- (1) virtual simulation for formation purposes before to perform the process (i.e the sequence of tasks) in the real world;
- (2) assistance while performing the actual maintenance procedure
 - (2.1) based on a pre-established scenario;
 - (2.2) based on the live support of a remote expert.

In our work, we focus on collaboration in MR, thus we contribute to fields (1) and (2.2) that intrinsically imply collaboration between an expert and an operator.

This applicative context and the associated example introduce several interesting topics that remain as open issues in the scientific community.

First, **asymmetrical aspects** can be found in a few dimensions of this kind of remote collaborative MR. In the previous example, roles of collaborators are different (expert/operator), thus they do not need the same requirements and will not act in the same way because of role differentiation. Thus, devices used could be heterogeneous, which is another asymmetrical dimension present in our context. Furthermore, remote collaborators share the same virtual world and are able to control their own viewpoint in it. Particularly, a last asymmetrical dimension is introduced if we consider that one of the collaborators (the expert in this case) can change its scale in the shared virtual world. This creates a multi-scale shared virtual world in which each collaborator can benefit from a suited point of view according to his/her task, role and devices.

Second, remote helping for maintenance implies different kinds of tasks. Indeed, to achieve the actual **remote guidance to help an operator performing in the real world**, the expert must be able to:

- guide the operator in navigation to place the operator in the correct location;
- assist the operator in selection/manipulation to perform the actual physical task;
- perform gestures guiding in order to explicitly explain to the operator the gestures to follow.

A highly connected field is **collaborative virtual simulation performed before acting in the real world**. Virtual Reality (VR) provides efficient collaborative systems to handle virtual simulation with remote users. In this context, **remote co-manipulation of virtual object** is still a challenging task to perform in VR without tangible link between collaborators. Indeed, in the real world, collaborative manipulation can only occur if collaborators are physically collocated (unless they use remote controlled robot), and the physical link created by the manipulated object is essential in the completion of this kind of task. In VR remote collaboration, this physical link is lost (except in case of bidirectional haptic feedback between the users), and thus these VR systems must provide other cues and tricks to overcome this loss in the collaborative process.

Third and last, even if MR systems provide advanced interactive worlds, nevertheless they suffer from issues that must be taken into account to provide acceptable and efficient applications. One of the strongest issues is linked to awareness [Gutwin and Greenberg, 1996]. In the online Cambridge dictionary, **awareness** is defined as ¹:

“Knowledge that something exists, or understanding of a situation or subject at the present time based on information or experience.”

Thus, when designing a MR application, it is essential to focus on this awareness aspect in order that the user understands what happens in the VE, either concerning his own presence into the VE, or what interactions are possible as well as their consequences on virtual objects. Moreover, when dealing with multi-user MR applications, and especially if collaborators are remote, collaborative awareness has a great impact on the collaboration effectiveness. First, the system must let know results of shared interactions into the VE in order to extend the single-user interaction feedback to multi-user settings. Second, the understanding of others' perception and activity help to disambiguate collaborative situations in a remote context where users can not see each other directly contrary to a physically collocated collaboration.

1. <http://dictionary.cambridge.org/fr/dictionnaire/anglais/awareness>, a more specific definition focused on our context is developed in chapter 2.

Objectives

The objective of this work is to improve remote collaboration in MR, especially in asymmetric context and leveraging awareness through visual feedbacks. To do so, we want to propose innovative concepts to perform remote assistance using the applicative context of industrial maintenance. Moreover, collaborative virtual simulation is also interesting in this context, because it is a way to train operators to perform tasks, without any danger and with reproducibility and monitoring capabilities, before to reproduce them in the real world. Thus, we defined several goals for our global approach:

1. We must take care of awareness issues in every proposals we do in order to provide efficient remote collaboration. Indeed, this is the key for a good understanding between remote users.
2. We must propose innovative systems to perform remote support for maintenance through MR technologies.
3. We must propose new ways to collaborate synchronously in virtual simulation, more precisely to co-manipulate shared content, in order to provide formation facilities adapted to our context.

Dissertation Organization

We just introduced our research context as well as challenges we are interested in this work.

Chapter 1 presents a global state of the art of domains implied into our work, that are synchronous remote collaboration as well as interaction techniques in MR and frameworks that manage this. Then, each next chapter (i.e 2, 3 and 4) contains a specific state of the art dedicated to its research topic.

Chapter 2 presents our first contribution that is a new model to define awareness in MR, either for single-user application as well as for remote collaboration. Then, we present an early user study that we have conducted in order to evaluate strategies of users to estimate the activity of a remote user according to available visual feedbacks and personal awareness. This gives us interesting results to focus in the following of our work concerning awareness in asymmetric remote collaboration context.

Chapter 3 focuses on managing actual remote helping, especially for maintenance purposes, in terms of navigation and selection/manipulation. We present prototypes and interaction techniques that we developed within two different approaches in this asymmetrical roles context: expert and operator. The first one is a multi-scale VE in which the remote expert has a global view of the operator's workspace. The second one is especially interesting because of its possibilities to manage remote gestures guiding and a better perception of the more natural remote helping cues by the operator. Both system have been evaluated with pilot user studies.

Chapter 4 focuses on collaborative virtual simulation by providing new ways to co-manipulate virtual objects. Thus, we propose new approaches for co-manipulation that handles every asymmetric dimensions we presented in the introduction, that are roles, scale and devices. The first technique is particularly interesting to handle asymmetric scale and devices in a generic way. The second one takes place in a multi-scale VE and is based on asymmetric roles. It goes further in the virtual co-manipulation concept by manipulating an object and a remote PoV at the same time. This approach is efficient because of the multi-scale approach we propose. More-

INTRODUCTION

over, the hybrid virtual object/remote PoV manipulation is an interesting concept to go further our previous work on remote navigation guiding.

Chapter 4.3.7 concludes this PhD thesis and gives some perspectives of this work.

Chapter 1

Related Work

1.1 Synchronous Remote Collaboration

In this section, we present asymmetry dimensions involved by remote collaboration in MR. Scheme 1.1 illustrates these dimensions. We divided this section according to these dimensions. First, section 1.1.1 presents some of these asymmetrical dimensions that are the points of view and the roles as well as an additional one that is the input and output (I/O) devices used. This third asymmetrical dimension is introduced because remote collaborators can use heterogeneous devices; moreover, this dimension can be split in terms of interaction devices and feedback devices. Then, section 1.1.2 develops fields of remote collaboration and especially focuses on MR that includes another asymmetrical aspect in remote collaboration that is the level of virtuality induced by the I/O devices used.

1.1.1 Asymmetry Dimensions

Dealing with remote helping through distant collaboration often leads to introduce asymmetrical dimensions in the collaborative process and/or setup. In our work, we focus on this asymmetric aspect in remote collaboration and more particularly to three specific dimensions. First, roles of collaborators can be asymmetric according to the collaborative scenario. Second, their Point of View (PoV) on the shared world and their perception can be different if they belong to a multi-scale shared environment. Last, the devices used can be heterogeneous, especially in a remote helping context. Even if they can be linked, we distinguish interactions spaces (issued from user's actions) and feedback spaces (used for user's perception) to be able to describe more specific situations. As an example, a user can have a visual feedback that merges real and virtual content while interacting on a real object or on a virtual one. Thus it can be useful to separate interactions on real objects from interactions on virtual ones. In the same way we can classify displays for visual feedback in terms of virtual or real content.

1.1.1.1 Roles

Different domains are interested in using remote collaboration in order to train a user or to help him in performing a specific physical task. Virtual simulations can be used to learn about procedures before to perform them in the real world, for instance in the industrial domain [Lopez et al., 2013], or in a medical context for surgery procedure teaching [Claude et al., 2015]. In this context, the trainer can remotely explain and show to one or more trainees the different steps

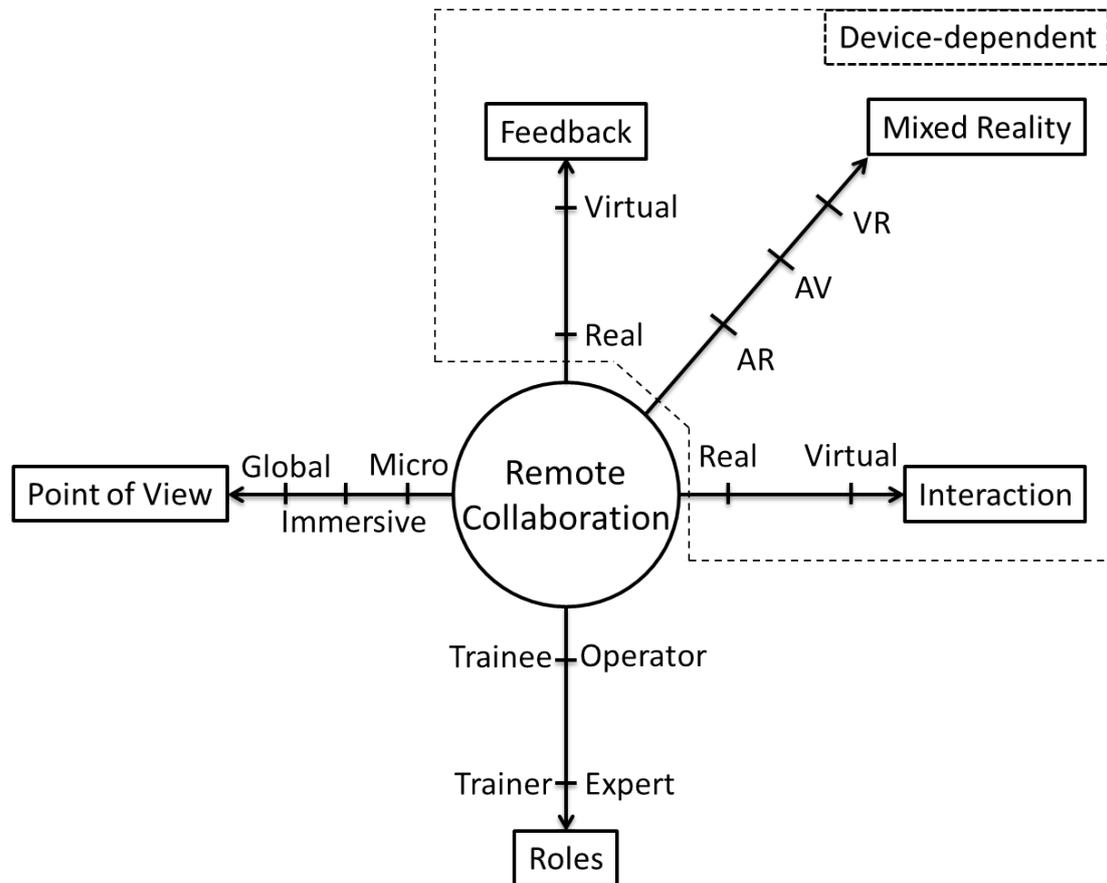


Figure 1.1: Scheme of the main asymmetrical dimensions found in remote collaboration in MR. We distinguish roles, point of view, and partially device-dependent dimensions that are interaction device and feedback device as well as the position of the application on the MR continuum, called level of virtuality of the MR application.



Figure 1.2: Illustrations of remote collaboration with asymmetric roles. (Left) A remote assistance system with expert and operator [Huang et al., 2013a]. (Right) A collaborative training application in the military domain with trainer and trainee [Lopez et al., 2013].

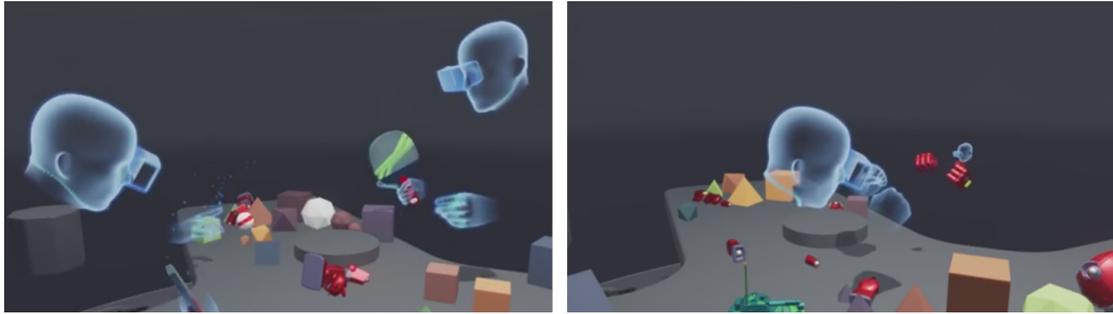


Figure 1.3: Illustrations of a sandbox providing collaborative remote interactions in a shared world, extracted from Toybox demo by OculusVR. On the left, both users interact at the same scale, whereas, on the right, a user is interacting at a lower scale than the other user.

to perform before to let them try in their shared virtual environment, under his supervision (cf. Fig. 1.2). In the same way, remote helping in performing a physical task also introduced asymmetric roles between users. Indeed, in this scenario, one user, considered as the expert, remotely helps the operator in performing a complex and specific procedure, unknown from the operator. For instance, it has been used in the mining industry [Huang et al., 2013a], or in assembly scenario context [Huang et al., 2013b, Robert et al., 2013] with remote guiding through hands gestures (cf. Fig. 1.2).

1.1.1.2 Point of View / Scale

In remote collaboration, users virtually share a common world while they are geographically far away. In general, they are able to control their individual PoVs. This aspect introduces the notion of multi-scale collaboration. Indeed, in some cases, users can be part of the virtually shared world at the same scale, but in other cases, it can be useful to let a user have another PoV on the shared world by changing its scale in it. It can be used to take advantage of a global view by increasing the PoV scale in order, for example, to guide another user in virtual exploration scenarios [Nguyen et al., 2012], or for architectural laying-out of spaces [Goldschwendt et al., 2014]. On the contrary, for entertaining purposes (cf. Fig. 1.3), or even for scientific data exploration (e.g microscopic data such as molecules), it can be interesting to be able to minimize someone's PoV in order to be able to perceive tiny details in the shared world. If only one user changes its scale, it introduces an asymmetric collaboration in terms of PoV. It leads to enhanced collaborative capabilities provided by different PoVs because users are able to communicate their own perception of the multi-scale shared world.

1.1.1.3 Devices

More and more devices become consumer-ready and are put on the market. Displays are more affordable, and consumer Head-Mounted Display (HMD) are released. But there are many manufacturers and proprietary devices that need to be normalized in order to be able to switch between any of them in a transparent way. This capability to adapt automatically the application to any devices is called interfaces plasticity [Lacoste et al., 2015a]. The same way, interaction devices (e.g controllers) are very heterogeneous whereas they provide equivalent capabilities (e.g buttons, joysticks and 6DoFs tracking). Figure 1.4 illustrates devices on the consumer market. Moreover, other kinds of devices provide different capabilities such as force feedback, naviga-

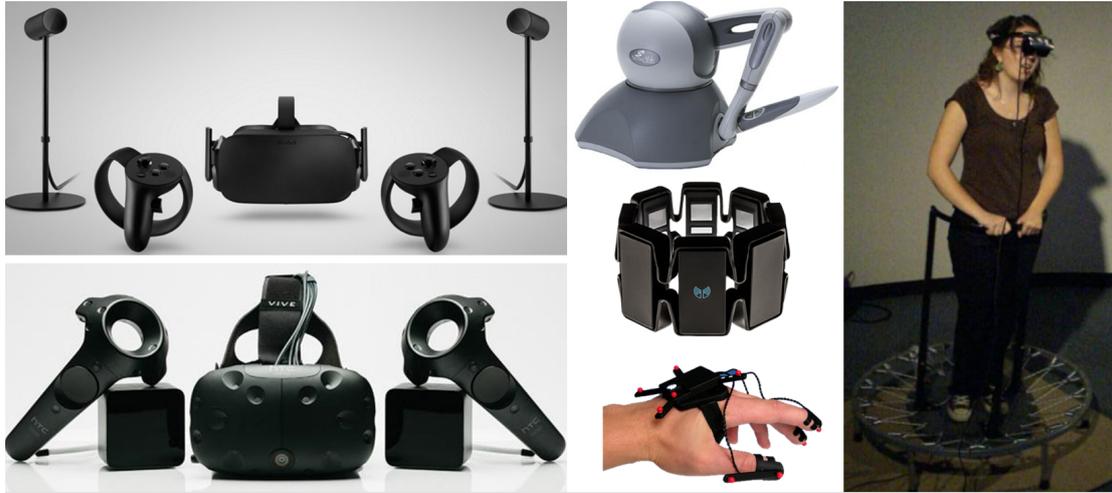


Figure 1.4: (Left) The Oculus Rift CV1 and the HTC/Valve Pre Vive, with their controllers and tracking system, available on the consumer market. (Middle) The Omni PHANTOM desktop haptic arm, the Myo device for hand tracking, the finger tracking system based on ART. (Right) the Joyman device [Marchal et al., 2011] for navigation.

tion or even hand tracking (cf. Fig. 1.4). This variety of devices introduces an asymmetric aspect in collaboration because users often use different devices, either because of their availability, or because of some needed capabilities.

1.1.2 Fields of Remote Collaboration

Providing multi-user systems allows to take advantage of cross-skills from each collaborator. In this context, several fields are interesting in remote collaboration. Historically, Computer-Supported Cooperative Work (CSCW) has been the first research field that has focused on remote collaborative through computers. It uses desktop computers in order to provide teleconferencing systems [Rae et al., 2015] as well as concurrent document edition while maintaining consistency in a synchronous or an asynchronous way [Rama and Bishop, 2006]. They also handle remote collaboration on multimedia content as well as 3D content [Casera and Kropf, 2010], but always using desktop computer and interaction devices (e.g mouse and keyboard / touchscreen). In this related work, we do not discuss about CSCW, rather we focus on Mixed Reality systems that provide immersive remote collaboration capabilities taking advantage of 3D technologies and interactions.

1.1.2.1 Mixed Reality Overview: VR-AR Continuum

Mixed Reality (MR) is a promising research area that proposes to mix real world with virtual artifacts. It offers natural ways to display virtual content taking advantage of real world referencing in order to ease interactions. This leads to more immersive and intuitive systems, and thus improves user performances. According to the Milgram's classification (cf. Fig. 1.5), Augmented Reality (AR) combines virtual objects into the real world, whereas Augmented Virtuality (AV) adds real items into Virtual Environments (VE). On both extrema are real world and Virtual Reality (VR) that is purely virtual without integration of real world's items.

The place an application is on the MR continuum defines its degree of virtuality, from purely real to purely virtual. Notice that an application can dynamically moves on this continuum at runtime, for instance from VR to AR [Billinghurst et al., 2001]. Moreover, **collaboration in MR introduces an additional asymmetric dimension that is the degree of virtuality for each user** that can run different kinds of MR applications while sharing a common synchronized virtual content.

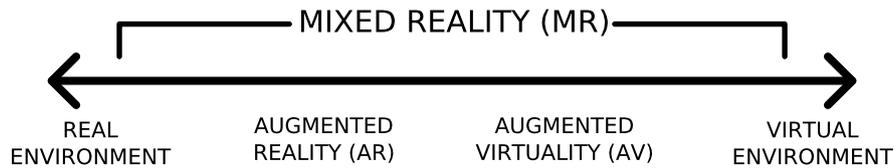


Figure 1.5: Milgram's continuum [Milgram and Kishino, 1994].

1.1.2.2 VR

To begin, let us define VR using a citation from Fuchs et al. [Fuchs et al., 2011] translated from the original version by Arnaldi et al. [Arnaldi et al., 2006]:

“Virtual Reality is a scientific and technical domain that uses computer science and behavioral interfaces to simulate in a virtual world the behavior of 3D entities which interact in real time with each other and with one or more users in pseudo-natural immersion via sensori-motor channels”

Immersion is a key element of VR. According to Sutherland, a VR pioneer, *“the ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.”* [Sutherland, 1965]

Existing VR systems can be classified in two categories according to the display system used: [Boas, 2013]

- semi-immersive: workbench [Poston and Serra, 1994], large 3DTV;
- immersive: immersive cube (e.g CAVE [Cruz-Neira et al., 1993]), HMD [Sutherland, 1968].

Figure 1.6 illustrates these kinds of VR systems.

In the context of remote collaboration, VR is a powerful tool in order to erase the physical distance between collaborators. Collaborative Virtual Environments (CVEs) provide shared VE allowing immersion for multiple, potentially remote, users and as natural as possible interactions capabilities in order to improve the collaborative process. Many applications can benefit from these features. Scientific visualization [Duval et al., 2008] can be improved by providing better perception abilities and also allowing a multi-scale visualization approach. Immersive teleconferencing system enhance presence feeling and non-verbal communication between distant users [Beck et al., 2013]. Likewise, training, virtual prototyping or even architectural exploration are also great fields of application for CVE.

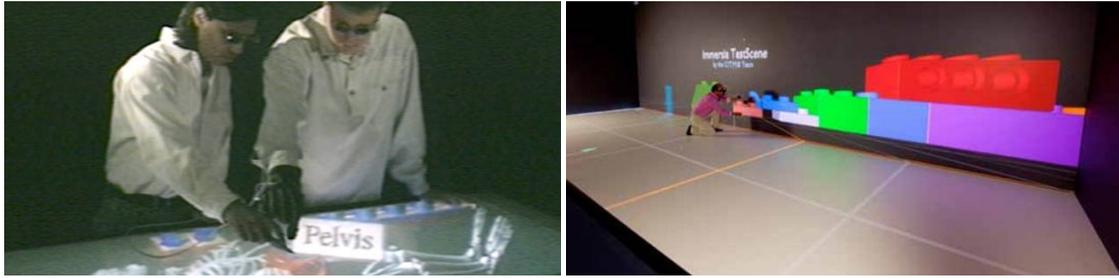


Figure 1.6: VR systems: (Left) a workbench semi-immersive VR system with two users collaborating [Agrawala et al., 1997]; (Right) Immersia, an immersive cube. Both systems provide 3D stereoscopy with head-tracking for user motion-dependent rendering.

1.1.2.3 AR

AR relies on the real world and combines computer-generated contents with the user perception of the real world. From Azuma et al. [Azuma et al., 2001], an AR application must follow these requirements:

- “combines real and virtual objects in a real environment;
- runs interactively, and in real time; and
- registers (aligns) real and virtual objects with each other.”

AR display systems can be classified in three categories:

- spatial AR: projection or on a fixed display;
- handheld: on a tablet or a smartphone;
- head-worn: HMD using virtual retinal displays (video-based or see-through) or contact lenses.

In all cases, AR faces three main challenges. First, initial registration and tracking of the user PoV in order to adapt the rendering in real time with a minimum end-to-end delay. Second, the consistency of the combination of the real and virtual content. Indeed, since virtual content is combined with the real world, it needs to be integrated in a proper way in order to be perceived correctly. Particularly, the handling of virtual lighting and shadows are challenging, as well as the occlusion between real and virtual content that is an important visual depth cue. Third, the display device itself is a technological challenge. It must satisfy several requirements: it must be comfortable to use (especially if it is Head-Worn) and must provide a proper perception of the augmented world. This last requirement is dependent, amongst others, of the display form factor as well as its FoV that needs to be large (as close as possible as the natural human FoV, e.g $\sim 200^\circ$ in binocular vision for horizontal FoV, with a $\sim 120^\circ$ binocular overlap [Dagnelie, 2011]).

Collaborative AR is mainly used with physically collocated users (e.g not remote, cf. Fig. 1.7), for instance in a face-to-face setup [Billinghurst et al., 2002b]. In a remote context, it can also be used for remote help in performing a physical task in the real world, while being helped by a remote user with 3D augmentations [Robert et al., 2013], or even for teleconferencing [Billinghurst et al., 2002a] overlaid in the real world.

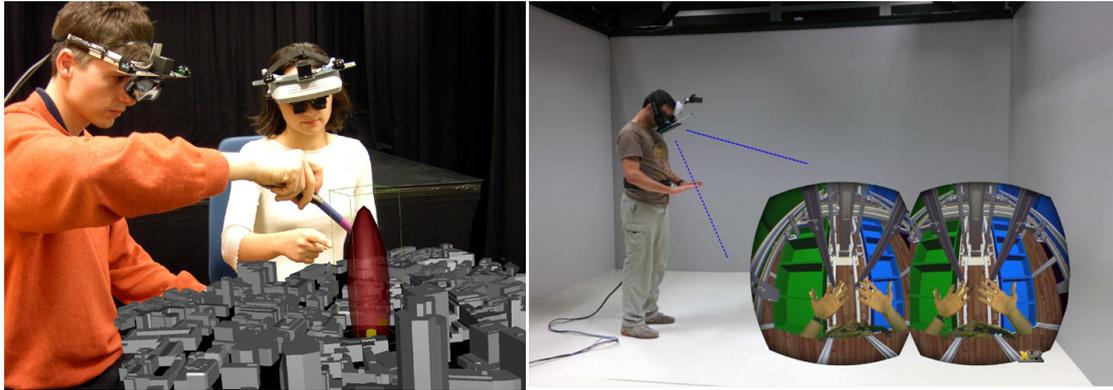


Figure 1.7: (Left) Illustration of ARTHUR, a collaborative AR system [Broll et al., 2004]. (Right) An AV simulation that includes real hands captured in 3D into the VE.

1.1.2.4 AV

AV takes place between VR and AR. This is a hybrid form of VR, that adds real content into a VE. For instance, it can be used to add real furnitures of our physical space, as demonstrated by Nahon et al. [Nahon et al., 2015], or even his own body (cf. Fig. 1.7) in order to be aware of it while being immerse into a VE. In remote collaboration, it is particularly interesting because we can add remote users into the shared VE in order to improve communication and mutual understanding of the other interactions. It is achieved using real-time 3D reconstruction technology and live streaming over the network [Petit et al., 2009].

1.2 Interaction Techniques

We have just presented the different modes of mixed reality to immerse a user in virtual content, mixed with a perception of the real world or not. But this immersion does not only consist in visualizing the worlds (virtual or real) but in general requires interactions of the users with contents. To achieve it, interaction techniques provide the ability to interact with an interactive system. It is the input link between the user and the simulation.

1.2.1 Classification

In MR, interactions tend to be the most natural as possible in order to decrease the learning curve thanks to real-world-like interactions. Thus they involve the user engagement into the interaction process by its motion and gestures. According to Bowman et al. [Bowman et al., 2004], 3D interaction techniques can be classified in 4 categories:

- navigation: to move in the VE;
- selection: to select an item;
- manipulation: to modify an item;
- application control: to change application parameters.

Navigation allows to move the virtual camera in order to choose the desired PoV into the VE. It includes the scale of the PoV that enables multi-scale VE. Selection and manipulation are used to interact and modify virtual items into the VE, while application control is needed to edit parameters as in classical desktop applications. This fourth category can be implemented based



Figure 1.8: Illustrations of mid-air bimanual interaction schemes in semi-immersive systems. (Left) A virtual carousel with symmetric 3D object manipulation inspired from work presented by Gaucher et al. [Gaucher et al., 2013]. (Right) A virtual design application [Le Chénéchal et al., 2014b] based on an asymmetric control scheme.

on desktop-like WIMP-fashion Graphical User Interface (GUI) adapted in 3D [Andujar et al., 2006], or it can be based on natural user input such as gestures [Jacob et al., 2008].

1.2.2 Interaction Metaphors

Even if they wish to mimic realistic interactions, interaction with virtual content are not exactly the same as in the real world with real objects. Thus, interaction techniques need to provide a comprehensive way to behave in order to be usable. The behavior that links user physical interaction with the corresponding action into the VE is called interaction metaphor. These metaphors can be classified in 3 categories:

- symbolic;
- pseudo-natural;
- natural.

For instance, navigation can be achieved based on a symbolic metaphor using a joystick, or a pseudo-natural one using body-based navigation [Marchal et al., 2011] or walking-in place technique [Usuh et al., 1999], or a natural one using walking as in the real world with possible redirection tricks to increase the virtual walkable area compared with the real tracking area [Razzaque et al., 2001].

Concerning selection and manipulation, the user's interaction tool can be a virtual hand or a 3D ray, for instance. An additional behavior can complete the metaphor to add capabilities. As an example, selection of far objects can be achieved using the GoGo technique [Poupyrev et al., 1996] that, basically, stretches virtual arms, at a certain speed according to the distance of the hand from the user, until casting an object. It is a derived version of the classical 3D ray metaphor that cast objects continuously. Hybrid approaches can be proposed, such as the HOMER metaphor that combines raycast for selection and virtual hands for manipulation [Bowman and Hodges, 1997].

1.2.3 Interaction Scheme

MR systems are basically real-time interactive systems that combine many inputs and provide multi-modal outputs (e.g leveraging different sensory channels) simulated according to the VE interactions. To combine several inputs of a single user in order to achieve interactions into the VE, MR systems propose interaction schemes. An interaction scheme combines user inter-

actions, through the devices used, with the interaction metaphors and the virtual object behavior in order to provide advanced interaction techniques.

For instance, using both hands to interact (cf. Fig. 1.8), an interactive system, and particularly a MR system, can handle this bimanual interaction in many ways in order to modify the state of the VE. For manipulation, it can be based on asymmetry between roles of dominant and non-dominant hand used as a frame of reference [Hinckley et al., 1998]. Another asymmetric approach consists in using the non-dominant hand for application control using gestures as commands, and the dominant hand for direct interaction in the VE [Le Chénéchal et al., 2014b]. Another way to handle bimanual interaction is to provide a symmetric interaction scheme, such as the *grab-and-twirl* or the *slide-and-turn* [Cutler et al., 1997], where the manipulation results from the average motion between both hands.

Finally, the inputs can also be multi-modal. For instance it can combine speech recognition, motion-based and gaze-based interactions. This provides enhanced interaction techniques because it allows to improve interaction performances by combining different human skills. For instance, selection task can take advantages of the speed of gaze motion and the accuracy of hand gestures [Stellmach and Dachsel, 2013].

1.2.4 Collaborative Interactions

Some tasks need more than one user acting at the same time on the same object to be achieved successfully. For example, some industrial manipulations [Aguerreche et al., 2010, Salzmann et al., 2009] cannot be done alone but need at least two collaborators to manipulate voluminous objects. In the same way, teaching scenarios involving a teacher and at least a learner [Luna et al., 2012] intrinsically require CVE. Figure 1.9 illustrates such collaborative MR applications. Moreover, other kinds of scenarios are performed better in a collaborative way. For instance, virtual exploration of large environment [Nguyen et al., 2013] can be enhanced by the help of a collaborator with a different point of view on the VE to guide the explorer.

In the literature several authors classified collaborative interactions. We present a classification based on a mix between the ones presented in [Margery et al., 1999] and [Luna et al., 2012]. It is a hierarchical classification from the basic features to the more advanced ones.

- *Level 1*: Awareness of team members' presence (avatars and communication channels).
- *Level 2*: Individual interactions on the scene.
- *Level 3*: Collaborative/codependent interactions (at the same time).
 - *Level 3.1*: Modification of different attributes.
 - *Level 3.2*: Modification of the same attribute.

Level 1 is similar to teleconference systems because it does not include any interaction capabilities, but only communication channels through audio and video communication facilities. Level 2 is an extension of interactive systems for a single user to multi-users environment. However, they do not adapt their interaction techniques from one to multiple users. Only Level 3 takes advantage of the availability of multiple users to improve their interaction capabilities. It can be achieved in two ways: using a split of Degrees of Freedom (DoF) policy (Level 3.1) or using an aggregation mechanism on a concurrent DoF (Level 3.2). It is partially similar to interaction techniques handling multiple inputs for a single user, such as bimanual interaction techniques that can use either asymmetrical [Hinckley et al., 1998] or symmetrical [Cutler et al., 1997] interaction schemes. Here, DoFs are not only the spatial attributes of a virtual object (e.g position, orientation and size, aka TRS for Translation-Rotation-Scale) but also its color, physical parameters or even ownership or shape for example. Thus, splitting these DoFs, as Level 3.1, eases the



Figure 1.9: Collaborative MR systems. (Left) A shared white-board with remote user 3D reconstruction and sketching interaction [Zillner et al., 2014]. (Right) Two users co-manipulate a virtual object and avatars are used for awareness of others [García et al., 2008]

merging process between the different inputs of collaborators because each DoF is controlled by only one input. On the contrary, Level 3.2 handles concurrent and synchronous modification of the same DoF by different users. In this case, proposed interaction techniques must handle the merging of different and potentially divergent inputs while maintaining awareness of the collaborators' interactions in order to understand the behavior of the modified object. First, the merging part can be based on the weighted average result of multiple inputs, with factors for each users associated to their roles for instance. Second, the awareness of collaborators' activity is essential to provide an understandable interaction technique. In cooperative manipulation (co-manipulation) task, the physical link formed by the virtual manipulated object can be simulated in a physically collocated context using a physical prop [Aguerreche et al., 2010]. In a remote context, haptic devices can substitute this loss of physical link [Sallnäs et al., 2000], but visual feedbacks can also be used to maintain this awareness of others' interaction, such as the use of rubber bands [Aguerreche et al., 2009b] or bent rays [Riege et al., 2006] that create a visual continuity between the virtual manipulated object and the users that collaboratively moves it.

1.3 Framework for Remote Collaboration

Relying on the classification of collaborative interactions and on the asymmetrical dimensions we are interested in, we identified three categories to define frameworks handling remote collaboration in MR. First, section 1.3.1 presents frameworks built to manage collaboration at Level 2 that induces a synchronization of the shared world that is the minimum requirement for interactive remote collaboration. Then, section 1.3.2 introduces PoV and roles asymmetry that add the need to handle synchronization of multi-scale shared VE. Devices asymmetry is not used in our categorization because it is a basic feature managed by MR frameworks, even for single user systems. Last, section 1.3.3 details more advanced frameworks that also handle Level 3 with collaborative interactions management.

1.3.1 Shared World Synchronization

Shared world synchronization between remote places is based on massive communication capabilities over the network able to transmit individual interactions in the VE as well as virtual

objects states to remote users in order to maintain consistency into the shared VE. It implies maintaining object's ownership, transformation, appearance, and others due to interactions of a single user on the virtual object. Low-level frameworks are able to manage this by using optimized networked communications based on TCP or UDP protocols over LAN, WLAN or WAN facilities. They can adopt different software architecture according to the need of the application, e.g high consistency, low delay or failures recovery for instance. The two main architectures are client/server that centralizes the simulation on a master node that ensures the synchronization between the others connected nodes, and peer-to-peer (P2P) that distributes the simulation synchronization over all the connected nodes [Frey et al., 2008]. Hybrid architectures have also been proposed as in [Fleury et al., 2010b] that take advantages of both previous architecture while handling additional features such as nodes migration or proxy objects for higher consistency. Commercially available frameworks such as RakNet¹ or SmartFox Server² manage massively multi-users synchronization in real-time. Their offer high performances, but do not handle advanced features encountered in remote collaboration in MR. However, they can be the base for network communication used in middlewares specifically designed for remote collaboration in MR.

1.3.2 Multi-Scale

As soon as we are facing asymmetric roles or PoV into remote collaboration, the framework used must be able to manage synchronization between multi-scale shared VE. Indeed, it can not be managed only by moving the camera in the scene to modify the PoV on the scene, others considerations must be taken into account as explained by Wartell et al. [Wartell et al., 2009]. Typically in VR systems, the display handles head-tracked dependent rendering to provide a collocated display. This feature is also dependent of the physical screen size, and thus constraints the virtual frustum used to compute the scene rendering. In the case of a stereoscopic rendering, disparity between right and left images (one per eye), used to generate depth perception, must be set according to the scene scale. Indeed, at 1:1 scale, average inter-pupillary distance (IPD) is 6.5cm. If we keep this value when modifying the scene scale, it results in discomfort viewing condition and wrong depth perception. Particularly, in collaborative cases that mix several fields of MR, such as AR and VR, it can be tricky to achieve multi-scale synchronization because of different real-world referencing; as an example in this case, the real world in AR, and the screen size constraint in VR. Indeed, because these two physical dimensions can not be dynamically modified, if a user wants to change its PoV scale at runtime, the application must adapt its synchronization mechanism in order to keep a consistent rendering for each user (including the representation of others in this multi-scale shared world).

It can be achieved using scene-graph facility [Strauss and Carey, 1992], implemented in modern 3D engines such as Unity3D³ or the Unreal Engine⁴, taking advantage of its hierarchical construction to compute virtual object transformations. With this approach, synchronization inside a multi-scale shared world can be simply achieved with a main root node storing the local scale of the shared world for each users and a synchronizations of objects transformations based on local transformations (as opposed to global transformations that is not relative to its parent node transformation). As an example of a middleware that provides facilities to develop such

1. <http://www.jenkinssoftware.com>

2. <http://smartfoxserver.com>

3. <http://unity3d.com>

4. <https://www.unrealengine.com/what-is-unreal-engine-4>

multi-scale shared world, MiddleVR⁵ company proposes *improov*^{3 6} as a platform for facilitating virtual remote collaboration for professionals in such environments. Nevertheless, none of these commercially available tools natively manages collaborative interactions at Level 3, but are limited to Level 2 of collaboration.

1.3.3 Collaborative and Concurrent Interaction

Collaborative interactions are often application-specific, this is why there is not any available framework that provides a generic way to manage collaborative interactions. However, several academic projects aimed to develop such a framework providing some collaborative interactions capabilities.

One of the first proposed framework is DIVE [Frécon and Stenius, 1998] that has evolved since almost twenty years with several versions. FIVE [Slater and Wilbur, 1997] as well as MASSIVE [Greenhalgh and Benford, 1995] are also early frameworks that manage scalability of the CVE and particularly focus on users' representation and awareness issues. They natively manage some awareness features with an integrated model inspired from the aura/focus/nimbus model proposed by Benford et al. [Benford and Fahlén, 1993]. Therefore they do not address some asymmetric dimensions we need for our work such as highly heterogeneous devices or asymmetric MR setups.

More recently, the MORGAN framework [Ohlenburg et al., 2004] proposes to handle these missing features by providing a solution able to manage multi-user VR/AR environments. Moreover, they implement their framework in a 3D engine-agnostic way in order to be bindable for future 3D engines that quickly evolve with hardware evolution.

Other frameworks specifically focus on the coherency issue such as OpenMASK [Larroché et al., 2008] or Collaviz [Dupont et al., 2010]. They implement distribution models based on proxies and referents on a hybrid architecture mixing peer-to-peer and a centralized-server to manage synchronization between distant places while minimizing network delay and inconsistencies. Moreover, they natively provide facilitators for remote collaboration based on models developed in the literature such as the IIVC model [Duval et al., 2013] that ease understanding of interaction and perception capabilities of remote collaborators in order to decrease ambiguous situations. Furthermore, these frameworks have built-in modules that can manage co-manipulation interactions. However, even if their kernel is rendering engine-agnostic, their practical implementation is strongly based on existing 3D engines that are no longer up-to-date, respectively Ogre3D and JOGL/jReality. Thus, they can not easily be integrated into modern 3D engines such as Unity3D, Unreal Engine, Cry Engine or others that provide state of the art 3D rendering and lighting algorithms as well as cross-platform build capability.

Last, some frameworks propose to model relations between objects in order to describe a VE and its content as well as interaction capabilities, possibly collaborative, with interactive objects. Mascaret [Chevaillier et al., 2012] adds a semantic-based modeling engine in order to provide a high-level description language based on UML to design CVE. #FIVE [Bouville et al., 2015] provides abstractions and mechanisms that are able to model behaviors within a CVE through a relations engine and a collaborative interactions engine.

5. <http://www.middlevr.com>

6. <http://www.middlevr.com/improov>

1.3.4 Summary

We presented the main frameworks that have been developed to manage remote collaboration using MR technologies with concurrent interaction capabilities. Here, we summarize main features we need for our work and what the non-exhaustive listed frameworks are able to manage.

Framework	MR-enabled	Cross-platform	Customizability	Availability	Up-to-date
DIVE	No	No	Partially	No	No
MASSIVE	No	No	Partially	No	No
FIVE	No	No	Partially	No	No
MORGAN	Yes	No	Partially	No	No
OpenMASK	No	No	Yes	Yes	No
Collaviz	No	Partially	Yes	Yes	No
#FIVE	No	Yes	Yes	No	Yes
Mascaret	No	Partially	Yes	No	Yes

We notice that most of the proposed framework are focused on VR and do not manage MR applications. Moreover, few of them are really cross-platform in order to deploy applications on different Operating Systems (OS) such as Windows or Android for instance. And last but not least, most of these frameworks are not really up-to-date, what certainly explains that most of them are no longer available.

We found no available solution that fits all the requirements we need for our research. Thus, we introduced needed features into our own framework that we developed to manage asymmetric remote collaboration in MR that is presented in annexe A. Especially, we built our framework around a generic core module and then we wrapped it to use in the Unity3D engine that provides an up-to-date game engine, integrating among others a physics engine and advanced illumination algorithms, and that is a very popular game engine nowadays with an available free version.

1.4 State of the art analysis

We presented the main fields we investigated on during this PhD thesis related to remote collaboration in MR involving asymmetry dimensions and adapted interaction techniques. Section 1.4.1 sums up MR advantages and limitations, then subsection 1.4.2 discusses achievements and remaining challenges involved in remote collaboration.

1.4.1 Mixed Reality

The main advantage of MR technologies is the ability to provide natural interactions in an immersive environment that aims at enhancing performances based on more natural perception and interactions compared to more classical desktop setups. In particular, AR and AV are very interesting because they merge real world content with virtual one. This integration of real items into virtual simulation allows to propose innovative ways to handle remote collaboration taking into account real surroundings of the remote collaborators. Applied to remote maintenance, it is particularly useful because one of the collaborator (called the operator or the agent) must act in his real world in order to achieve a physical procedure while being guided by a remote expert. On the remote expert side, the main issue to deal with is to provide him with a good perception of the operator's workspace in order to be able to help him in a comprehensive way. Moreover, immersive interactions in a shared 3D space provide to the expert interesting capabilities in terms of guiding cues that can be combined with verbal or video channels. However, there are still several limitations induced by MR technologies. Firstly, devices have intrinsic limitations due to their hardware components. For instance, VR HMD have limited FoV and resolution, as well as see-through AR HMD. Moreover, tracking is a key component of any MR system. This feature has also physical limitations in terms of tracking area, accuracy, drift and calibration. Secondly, one of the most important remaining limitation in remote collaboration is the ability to maintain a correct awareness of the shared space as well as others' activity. Hardware limitations generate some of these awareness issues because they can not track precise fingers motion or facial expressions for instance, but specific feedbacks can be proposed in order to minimize this lack of awareness about the remote space and collaborators.

1.4.2 Remote Collaboration

Concerning remote collaboration, many work already managed to provide many users with some ways to collaborate effectively while being physically distant. The first research field on this topic is CSCW that handles collaborative work with virtual data managed by desktop computers. It is useful to collaboratively edit documents or to run project reviews for instance. In addition, research about telepresence adds the ability to communicate using video and audio channels. Some advanced systems even handles eye-contact for more natural discussions with remote people. Moreover, the video channel conveys interesting non-verbal informations about body language and facial expressions. However, available **interactions** for remote collaboration stay limited and do not allow collaboration to be as natural and efficient as local collaboration in the real world in the same room around the same table. The **perception** of distant activities is also a challenge to provide efficient collaborations. Interactions and perception in a context of remote maintenance are the two main challenges addressed in our work applied to remote maintenance. By enhancing the remote expert perception and interactions capabilities and by providing new ways to help the operator in achieving his physical task with well-integrated and easy-to-understand guiding cues, we plan to reduce the distance feeling in remote collaboration.

1.4.3 Challenges & Contributions Overview

In this section we develop the challenges we aim to tackle. They stem from limitations found in previous work that still need to provide more awareness of the shared and the distant activity to remote collaborators.

Figure 1.10 classifies our contributions according to the MR continuum and to the PoV symmetry or asymmetry. We cover the entire spectrum of the MR continuum with both symmetrical and asymmetrical PoV/scale in the shared VE. More asymmetrical dimensions (e.g roles and devices) are covered within each contribution.

First we propose a new model in order to describe an interactive MR system in terms of **user activity** that feeds the different inputs, and of **semantic feedbacks** considering remote collaboration, i.e if they convey information that maintains awareness about his/her self or others' activities. We also tested several ways to represent users in a multi-scale VE and ran an experiment to evaluate the interpretation of the others according to his own representation and activity. This leads to conclude about the need to take into account the expertise of the end-user in the design of collaborative MR applications; especially in our applicative use case, we must be aware that the operator will not have necessarily any expertise about the use of MR technologies, while the remote expert could be trained in using a more complex interface.

Then, we apply some findings about awareness issues to our remote guiding use case by proposing two different settings. Both are based on an operator using an AR interface, and a remote expert using a VR interface. Thus we add the additional level of virtuality asymmetrical dimension in this contribution allowing the expert to virtually share the real operator's workspace. This feature has been studied in order to overcome awareness limitations in such remote maintenance systems that can be found in previous work. Moreover, the expert's interaction capabilities are also proposed and allow him/her to visually help the operator in completing his maintenance process. They use natural, real-world-integrated, and not intrusive visual cues that improve the remote helping process in terms of understanding and completion time.

Last, in order to deal with asymmetric devices, roles and PoV in VR shared world, we introduce new ways to collaboratively interact in purely virtual simulation. Particularly, we extend classical co-manipulation technique to a hybrid paradigm that is close to a remote navigation technique that we just investigated.

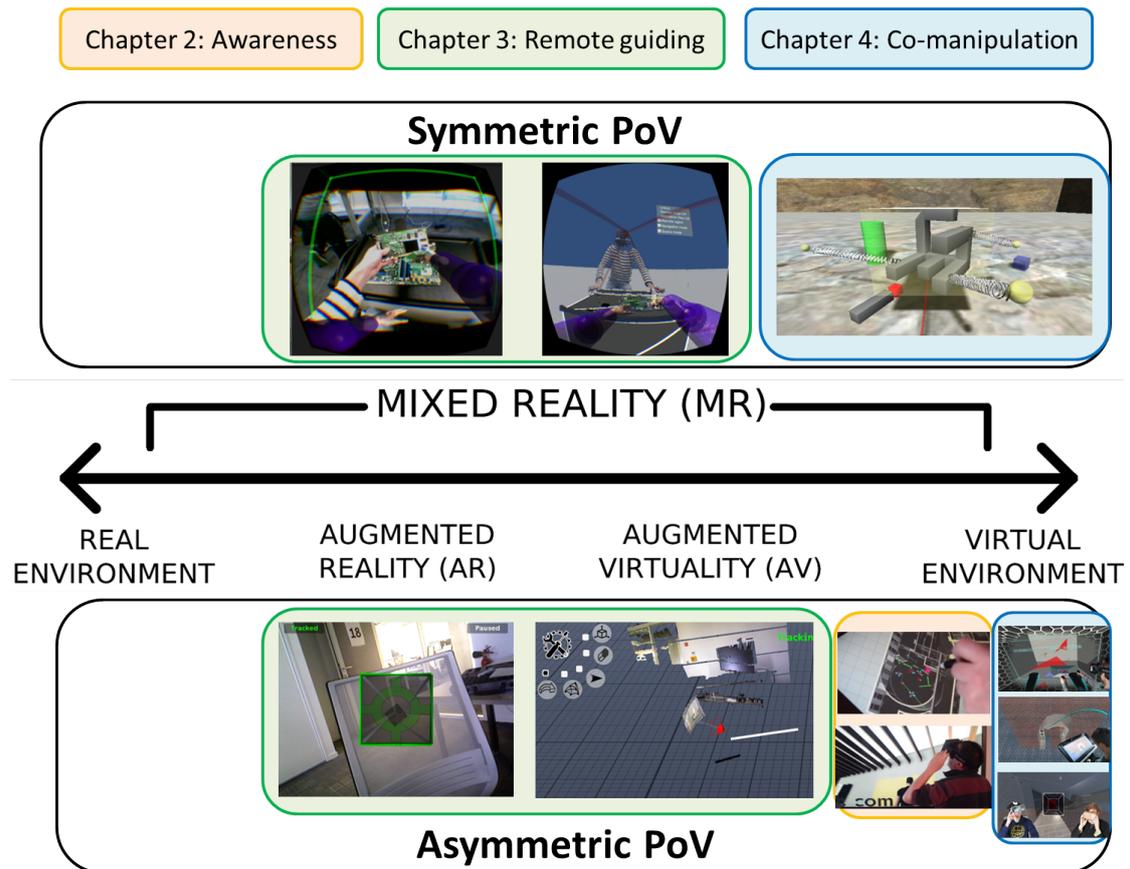


Figure 1.10: A summary of our contributions in three fields we focus on: awareness, co-manipulation and remote guiding. Contributions are classified according to their asymmetrical aspects in terms of PoV and level of virtuality. We can notice that asymmetrical dimensions in terms of devices and roles are not addressed in this figure.

Chapter 2

Awareness

Awareness is defined as the internal/mental representation of a user according to the interactive world he/she belongs to. This is a close definition of workspace awareness in CSCW context given by Gutwin et Greenberg [Gutwin and Greenberg, 2002]. In VR, it is mainly achieved through the interpretation of sensory feedbacks that inform the user about the consequences of his/her action, as well as the self-updating of the VE. In CVE, this awareness of the shared VE is more complex. Indeed, remote collaborators must interpret what is happening in the shared VE, which is also consequences of other's actions. Thus, collaborators' awareness involves interpretation of signals sent by others through the shared VE [Dourish and Bellotti, 1992].

This chapter begins with a state of the art of awareness in MR in section 2.1 with a focus on collaborative applications. Then, section 2.2 proposes a new model to classify users' inputs and feedbacks provided by the interactive system. These feedbacks can be generated from his/her own interactions into the VE or be issued from shared interactions with other collaborators. All these feedbacks are used in different ways in order to maintain awareness of the shared VE. Section 2.3 focuses on users' representation into a shared VE and develops some examples that demonstrate different levels of representation generating different levels of awareness in the user experience. In a similar way, section 2.4 discusses about PoV importance in remote collaboration in MR and gives examples of PoV impacts according to different awareness needs. Last, section 2.5 presents an early user study that we ran in order to address these two specific features that act on awareness in remote collaboration: users' representation and PoV. It gives us valuable preliminary results to our future work, especially focusing on asymmetry in remote collaboration in terms of roles and PoV.

2.1 State of the Art

Awareness is essential to let the shared VE be usable in potentially distant collaborative tasks. It allows users to perceive other actions and to understand the collaborators activities. Subsection 2.1.1 presents a classification of awareness features, then subsection 2.1.2 describes awareness solutions used to enhance the perception of the VE and its content. Subsection 2.1.3 develops some multi-modal awareness techniques. Last, subsection 2.1.4 describes some awareness solutions especially designed to enhance collaboration in CVE.

2.1.1 Awareness classification

Classically in VR, system interactivity is defined as a 3-step loop that is: perception, decision, action. Another way to express it follows an *interactive loop* defined in three steps: awareness, action, feedback [Goebbels et al., 2003]. In this last loop, the first step¹ indicates to the user what can be done in the VE, e.g what interactions are possible. The second step represents the actual interaction of the user with the VE, then the third step provides a feedback to let the user be aware of the consequences of his/her actions on the VE. Then, the loop can go on. Without it, there would be no mean for the user to directly understand his/her actions and the application would not be interactive enough. The lack of direct feedback drastically decreases the usability of the system. Goebbels et al. [Goebbels et al., 2003] proposed an extension of this *interactive loop* to be used in CVE. In particular, they extend the awareness part and divided it as follows:

1. proprioception, e.g perception of his own actions;
2. perception of his own physical and virtual input devices and of the virtual data set;
3. perception of co-presence, e.g of other collaborators;
4. perception of co-physical and virtual input devices and of shared virtual data set;
5. perception of co-knowledge and co-status.

These five steps are the ordered steps perceived by users at each step of the awareness part of the *interactive loop* defined previously. We note that steps 1 and 2 are the same in a single-user or a multi-user setup, and steps 3 to 5 specifically address awareness features in collaborative settings.

The term *presence* defines "*the feeling of existence within a given environment*" [Zahorik and Jenison, 1998]. Sanchez et al. [Sanchez-Vives and Slater, 2004] claim that *presence* is a level of consciousness above awareness. *Co-presence* [Goebbels et al., 2003] extends this consciousness feeling for collaborators in shared worlds. Thus, the last 3 steps enable coordination between users and improve collaborative interactions by trying to compensate the lack of direct collaborators' proprioception feedback.

Awareness is not only studied by computer science researchers, it is inspired by work from the social science community, and adapted to the context of CSCW and CVE. Social science community provides an exhaustive classification of awareness, but CSCW and CVE do not require every features of this classification. A survey about similarities between awareness in CSCW and social sciences can be found in [Gross et al., 2005]. We rather present a specific classification of awareness for CVE extracted from [García et al., 2008] and [Pinho et al., 2002]. Thereby, awareness in CVE can be split into 6 classes:

1. *Object*: is it selectable? selected? who is its owner? can we modify some of its attributes?
2. *Task*: who is currently participating in the task? what is the task status?
3. *World*: where are the objects? the users?
4. *Group*: who is online? what are the group members actions? positions? capabilities?
5. *Social*: what are the available communication channels?
6. *System*: what are the available data about the system states (stability, network latency,...)?

1. The first step of the loop is named *awareness* in the literature. However, it only refers to the interaction capabilities awareness part of the whole awareness that the system can provide (that includes direct interaction feedbacks).

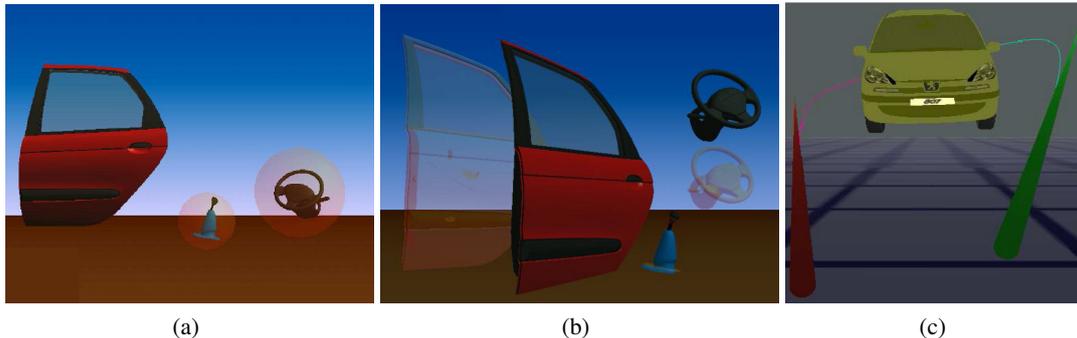


Figure 2.1: Illustrations of visual cues used to enhance awareness in CVE: (a) Bounding boxes indicating a specific object property [Duval et al., 2006b], (b) Echoes managing network troubles [Duval et al., 2006b], (c) Rays from the owners to the manipulated object [Aguerreche et al., 2009a].

This need of informations must be well-studied for each application and must not disturb the use of the system. For instance we must avoid to provide too many data that could hide some important parts of the VE. Rather, awareness must improve interaction performances by providing necessary visual, auditive or haptic cues together with a direct verbal communication channel at least. This must lead to enhance the whole perception of the VE including remote users, their current actions and their interaction capabilities.

2.1.2 Awareness basics: VE and content

First, VE must provide some ways to users to have an accurate perception of the whole environment. In immersive settings (e.g 1:1 scale), the World In Miniature (WIM) technique provides a way to have a global view of the whole VE in a map-like view. Without it, users only see what is contained in their respective view frustum. Most of the time in non-immersive setups, the available horizontal field of view (fov) is limited to 60° , whereas natural range of binocular vision extends up to 200° . Fraser et al. [Fraser et al., 1999] proposed peripheral lenses to provide such a wide horizontal fov. These peripheral lenses are placed on both sides of the central classical 60° fov and display another 60° fov on a thinner width. It provides peripheral blurred views mimicking natural peripheral vision. Users can momentarily switch the clear view and thus better perceive their environment and the collaborators' location and action.

Second, VE can let users be aware of the objects' interaction capabilities to improve their usability. For instance, interactive objects can display their current state by changing some of their appearance attributes when they are selected or manipulated. Another way to represent these state modifications consists in displaying a semi-transparent bounding-box (cf. figure 2.1a). Note that this method can also be extended using *echoes* in order to inform users about network troubles concerning shared virtual objects [Duval et al., 2006b] (cf. figure 2.1b). Moreover, when an object is selected or manipulated by a user named the owner, it is interesting for other users to know who is this owner. A typical technique consists in displaying the owner's name next to the interactive object, but more visual awareness techniques propose to display a ray (or an extended arm) from the owner location to the object (cf. figure 2.1c). Concerning interaction capabilities, objects can provide visual cues such as manipulation arrows in order to inform the users which interactions are possible.

Last, a famous approach considered two distinct spaces for each user named the *focus* and the *nimbus* [Benford and Fahlén, 1993]. In [Benford et al., 1994], *focus* and *nimbus* are "sub-spaces within which an object chooses to direct either its presence or its attention". More specifically, "the more an object is within your focus, the more aware you are of it", and "the more an object is within your nimbus, the more aware it is of you". It means that, in a generic way, interactive objects can react to user position and focus by changing appearances for example, and also inform others about user's attention and interaction capabilities.

2.1.3 Multi-modal awareness

A MR system can provide awareness through three different channels. First, visual feedback of the interactive tool allows the user to better anticipate the consequences of his actions. It is essential in immersive setups as in [Ortega and Coquillart, 2006] to improve the immersive feeling of the user in the VE. In this example, the user uses a physical prop to interact with the VE; the interactive tool is directly controlled by the prop motion and its visual representation is co-located with the tip of the prop in order to provide a convincing 3D user interface. Second, auditory feedback can enhance user perception of the VE. For instance, physical behaviors can be simulated thanks to specific sounds, such as contacts and friction in [Le Chénéchal et al., 2014b]. Third, haptic feedback improves kinaesthetic feedback and helps the user to coordinate his movements. In [Murayama et al., 2004], Murayama et al. used two SPIDAR devices, one for each hand. They provided separate haptic feedback for each hand and increased the bimanual interaction abilities thanks to the feeling of forces applied by hands in the VE. Other pseudo-haptic feedbacks such as proposed in [Sreng et al., 2006] can provide visual or auditory cues to overcome the lack of haptic device.

2.1.4 Awareness techniques for CVE

In the literature, many solutions have been implemented to enhance collaboration in CVE thanks to awareness. We do not present an exhaustive list of these solutions, rather we discuss some of them that are widely used and quite generic.

2.1.4.1 Awareness of collaborators

First, users have to be able to perceive their collaborators into the shared VE. To achieve it, the typical solution is to use an avatar animated according to the user motion. In the simplest case, only the body of the avatar is animated, but more advanced research worked on realistic avatar [Beck et al., 2013] textured as the real user, and even emotive avateering [Freeman et al., 2003, Niewiadomski et al., 2011] in order to enhance non-verbal communication between remote users. In this field, an interesting modality is gaze tracking that enables eye-contact and awareness of others' gaze [Steptoe et al., 2008]. Avatars with gaze direction allow fine and informative non-verbal communication and can reproduce natural social interactions.

Second, users should perceive others' interactions capabilities to be able to collaborate effectively. These capabilities can depend on hardware setups because users can interact with heterogeneous setups, such as an immersive visualization room with full body tracking, a semi-immersive 3D display with a 6DOF tracker input or a head-mounted display with tracked hands. Figure 2.2a gives an illustration of two heterogeneous setups. This heterogeneity leads to different interaction capabilities for each user. Collaboration becomes more effective if everyone is aware of these limitations. Duval et al. [Fleury et al., 2010a] proposed the Immersive and

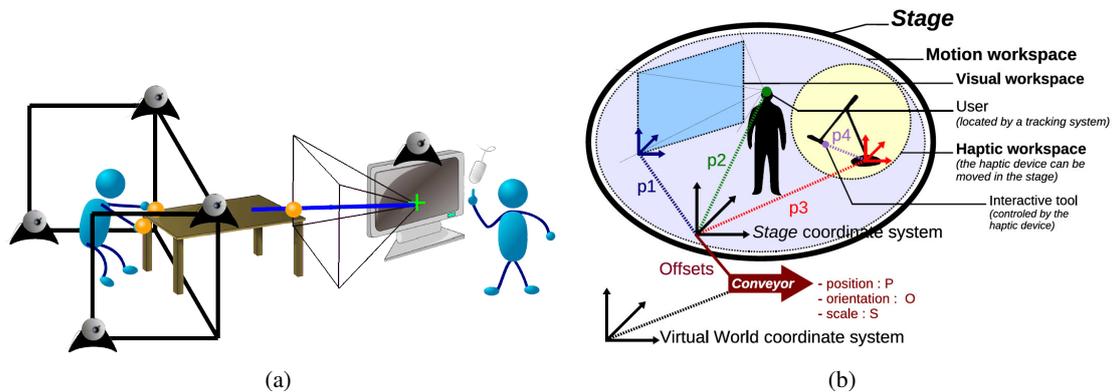


Figure 2.2: (a) An illustration of a collaborative session with two users. The one on the left works in an immersive setup, while the one on the right uses a desktop setup [Duval et al., 2013] (b) Explicative scheme of the different workspaces composing the IIVC and their relations [Fleury et al., 2010a].

Interactive Virtual Cabin (IIVC) as a generic concept to handle this awareness feature. It allows to define the real 3D volumes that represent different workspaces linked to a specific modality (e.g motion, visual, auditory,...) and to let other users perceive these spaces. These virtual spaces can stem from some tracking limitations or screen size, and to be aware of that avoids misunderstanding between collaborators, for example representing the view frustum [Fraser et al., 1999]. Figure 2.2b explicates the generic concept of the IIVC.

2.1.4.2 Collaborative awareness through communication

In this section, we present general awareness techniques providing users with ways to communicate. First, CVE must allow direct and natural communication. Several channels can be employed such as an audio channel. The audio communication has been proved to be the most important and essential for effective collaboration [Goebbels and Lalioti, 2001]. Video [Pauchet et al., 2007] or haptic [Chellali et al., 2010] channels can also be used. Second, symbolic communication is another way to allow communication between remote users. For instance, users can deposit arrows, compass or lighting into the VE to inform collaborators about interesting spots [Nguyen et al., 2013]. They can also create notes and post-it to add informations collected in the VE to share them with other collaborators.

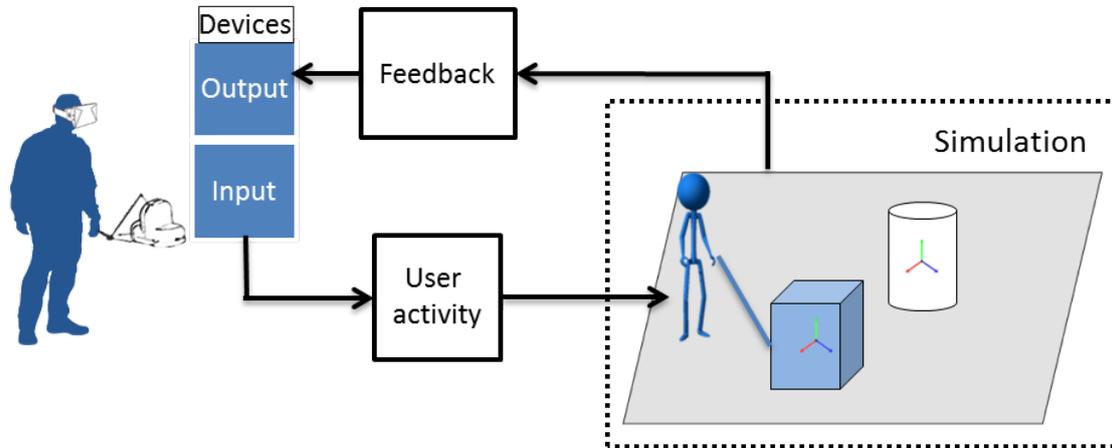


Figure 2.3: The classical input/simulation/output loop describing an interactive system.

2.2 A new generic awareness model

Many awareness techniques have been proposed, and some classifications allow to describe each of these techniques in terms of conveyed informations. Some models also provide generic ways to describe interactive and multi-user systems. However, awareness is very personal because it is generated by “*human perception and experience of events*” [Cai and Yu, 2014] according to provided feedbacks managed by the system. These feedbacks can also be classified, thus we propose a new model for VE that describes each types of available feedback. First, for a single user in section 2.2.1, then for multiple users in CVE in section 2.2.2. Last, section 2.2.3 adds the asymmetrical dimension in virtuality to our proposal in order to provide a formal way to model numerous kind of collaborative MR application. **In next chapters, each of our proposals instantiates this model.**

2.2.1 Single user

An interactive simulation is basically described as a 3-steps continuous loop:

1. input: user interaction;
2. simulation step: interactive application;
3. output: user perception.

Figure 2.3 illustrates this loop in our formalization. Physical input generated by user’s action is perceived as input, called *user activity*, for the interactive simulation. Then, the simulation generates *feedbacks* conveyed to the user through output devices.

In our detailed model (cf. Fig. 2.4), we split the input user interaction into two categories. We distinguish user activity depending on their nature. First, the user can change his state in terms of body posture (including head and hands for instance) in order to look around him/her in the VE or to take a comfortable posture in his/her real world he/she belongs to, or even reacting to an external stimuli from his/her real world. Second, the user can try to modify the VE content by explicitly interacting with it. The first input of user activity is called *user state*, while the second is called *interaction*.

Concerning the feedbacks provided by the VE, we split them into three categories in order to distinguish:

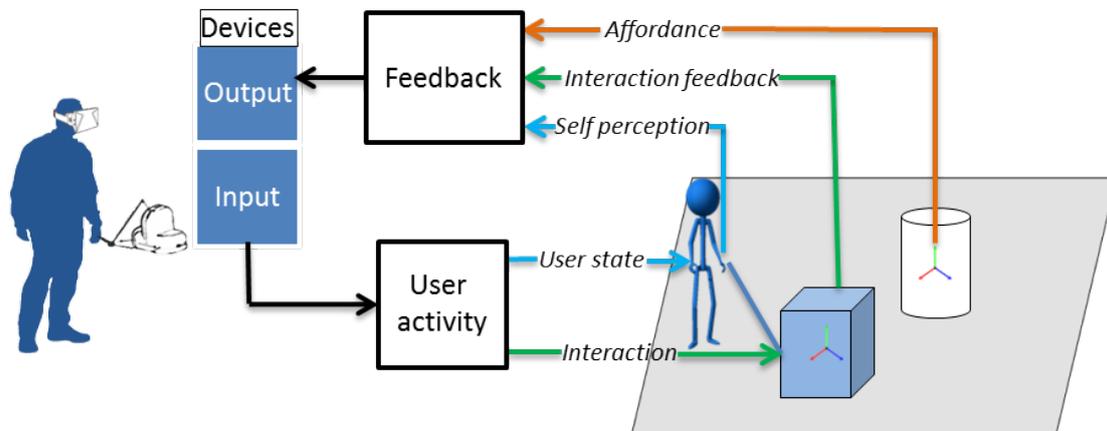


Figure 2.4: Interactive system model detailing types of user activity and feedback.

- feedbacks from intrinsic users properties (self-perception);
- feedbacks from intrinsic objects properties (affordance);
- feedbacks issued from the two previous ones: interaction between users and objects (interaction feedback).

First the *user state* can modify some content of the VE linked to the user him/herself. Indeed, the user can be represented by an avatar for instance, thus his state is directly controlled by the user state. The same way, interactive tools are often represented into the VE. These tools are controlled by user state, most of the time by user's hands motion, thus user's motion has consequences on his representation as well as his controlled interactive tools into the VE. We called this kind of feedback *self perception* because it is linked to user state who does not necessarily intend to interact with the VE content (except his/her own representation that we distinguish from the rest of the VE).

Second, VE can also provide informational feedback on possible interactions on different virtual contents. For instance, if a virtual object is movable, the object can carry manipulation arrows that give visual cues to the user about its interactive capabilities. This kind of feedback is called *affordance* [Gibson, 1979] of the VE that conveys informations to the user in a passive way (e.g without any explicit user's interaction) about interactive capabilities of its content.

Last, user's *interaction* on VE content has consequences on scene state which are computed by the interactive simulation at each simulation step. As examples, the selection of a virtual object can change its color, or its manipulation will move it according to the implementation in the simulation that can, for instance, include physical constraints. This feedback of the user's *interaction* is called *interaction feedback* and is the direct consequence of intended user's action on virtual content of the VE.

All these feedbacks are perceived by the user through his/her output devices. Then, the user builds an internal representation of the VE and his/her actions within it that is called *awareness*. Figure 2.5 integrates this awareness concept in our detailed model as well as the human cognitive loop known as perception/decision/action [Mallot, 1997]. From this point of view, awareness is generated by the combination of three elements in our model: feedbacks, output and perception. It starts from the simulation that generates feedbacks, then the output devices that convey these informations that are perceived by the user and used to maintain his personal awareness.

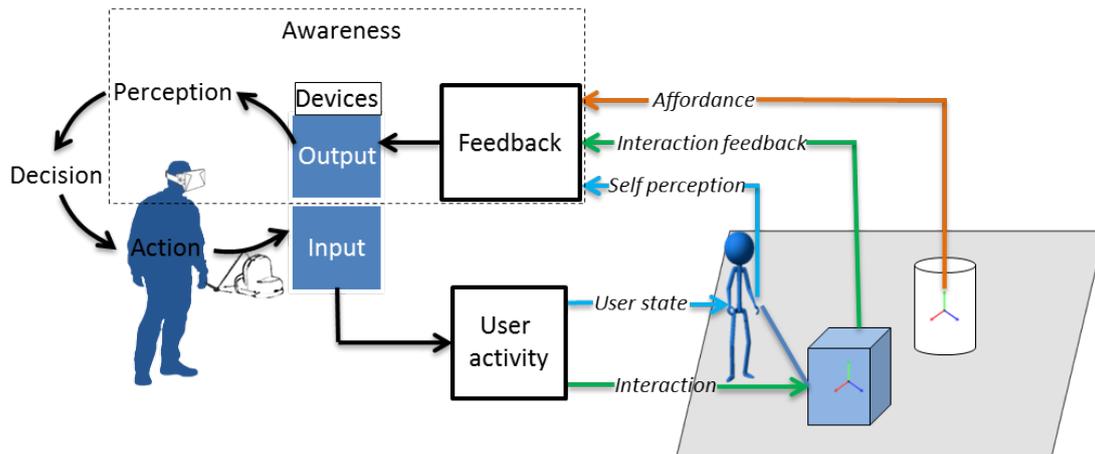


Figure 2.5: Interactive system model integrating user cognitive loop.

2.2.2 Collaborative setting

Figure 2.6 presents our model extended to multi-users shared worlds. When including additional users into the shared VE, the simulation step is also dependent of others' interactions. Thus, the interactions feedbacks are now consequences of potential *shared interactions* on the same virtual content.

Moreover, in order to address co-presence, the VE must also handle the representation of remote collaborators to be able to understand what others are doing. It is also useful for improving coordination with the ability to be aware of others' interaction capabilities; in the same way it is important to be aware of virtual content interactive capabilities (*affordance*). Here, we call *distant user perception* the feedbacks representing other users (including their interaction tools). It is an additional information that feeds the user's perception in order to maintain awareness of the VE shared with other users.

In next chapters, our work focuses on asymmetry in terms of roles, PoV, devices and level of virtuality. Thus, we do not specifically address *affordance* in shared VE. Rather, we focus on *self-perception*, *shared interaction feedback* and *distant user perception* that are the keys to improve awareness of one's self and others as well as shared activity when dealing with asymmetric remote collaboration in MR.

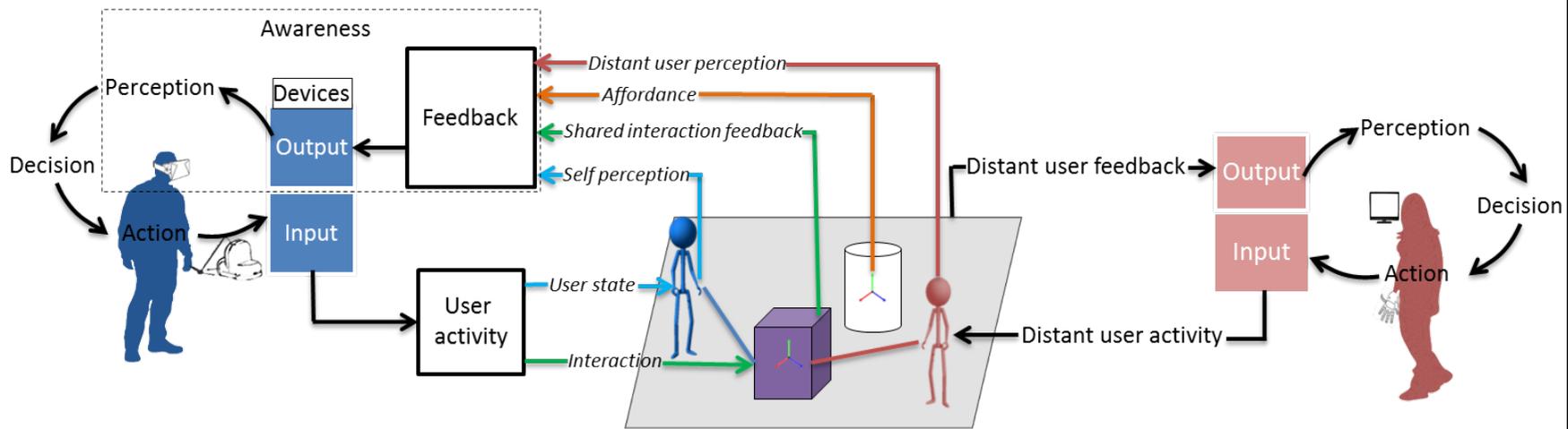


Figure 2.6: Our generic awareness model for CVE. We only detailed the blue user activity and feedback for clarity purpose.

2.2.2.1 Personal spaces and available feedbacks

We just presented a formal way to describe a CVE, in terms of users' input and feedbacks, considering symmetrical feedbacks (i.e all the users have the same capabilities in terms of perception on the shared VE, and feedbacks, especially visual feedbacks, are the same for every collaborator). However, in many situations this is not exactly the case. For instance, a common case is to let users interact in a multi-scale shared world. In this case, users will not require the same feedbacks, especially visual ones, because they do not perceive the CVE at the same scale. Thus, tiny details for a user with a global PoV are not relevant; instead, this user should be able to perceive global scheme that can not be seen at another scale.

Thus, in order to deal with these kinds of requirements, we propose to add the concept of personal spaces. It adds a layer to our model that separates real and virtual spaces, because each user has a personal virtual space in addition to his/her physical space. Figure 2.7 shows an example of this additional feature of our model in a multi-scale CVE scenario.

2.2.3 Multi-users MR setting

We proposed a model for CVE that handles many asymmetrical dimensions we introduced in our research. It can manage asymmetrical roles, devices and PoV. Still, the application level of virtuality for each user can also be asymmetrical: users can collaborate at different levels of virtuality while sharing a common VE. As an example, a user can use an AR application and another user a VR application while still collaborating on a shared synchronized VE. However, the synchronization of the VE depends of each user's level of virtuality.

Figure 2.8 illustrates our final model that takes into account this additional asymmetrical dimension in MR remote collaboration. Interactions in AR can be performed on real objects. In this case, the AR user can directly perceive the feedbacks of these interactions on real objects; these interactions generate a *real objects activity* that is used to synchronize virtual representations of real objects used to maintain awareness for others collaborators. A noticeable feature of this model is that aggregation of feedbacks from real and virtual space for each user results in equivalent perceptual abilities (merging virtual and real contents). Thus, collaborators are able to maintain equivalent levels of awareness and can collaborate in an understandable way.

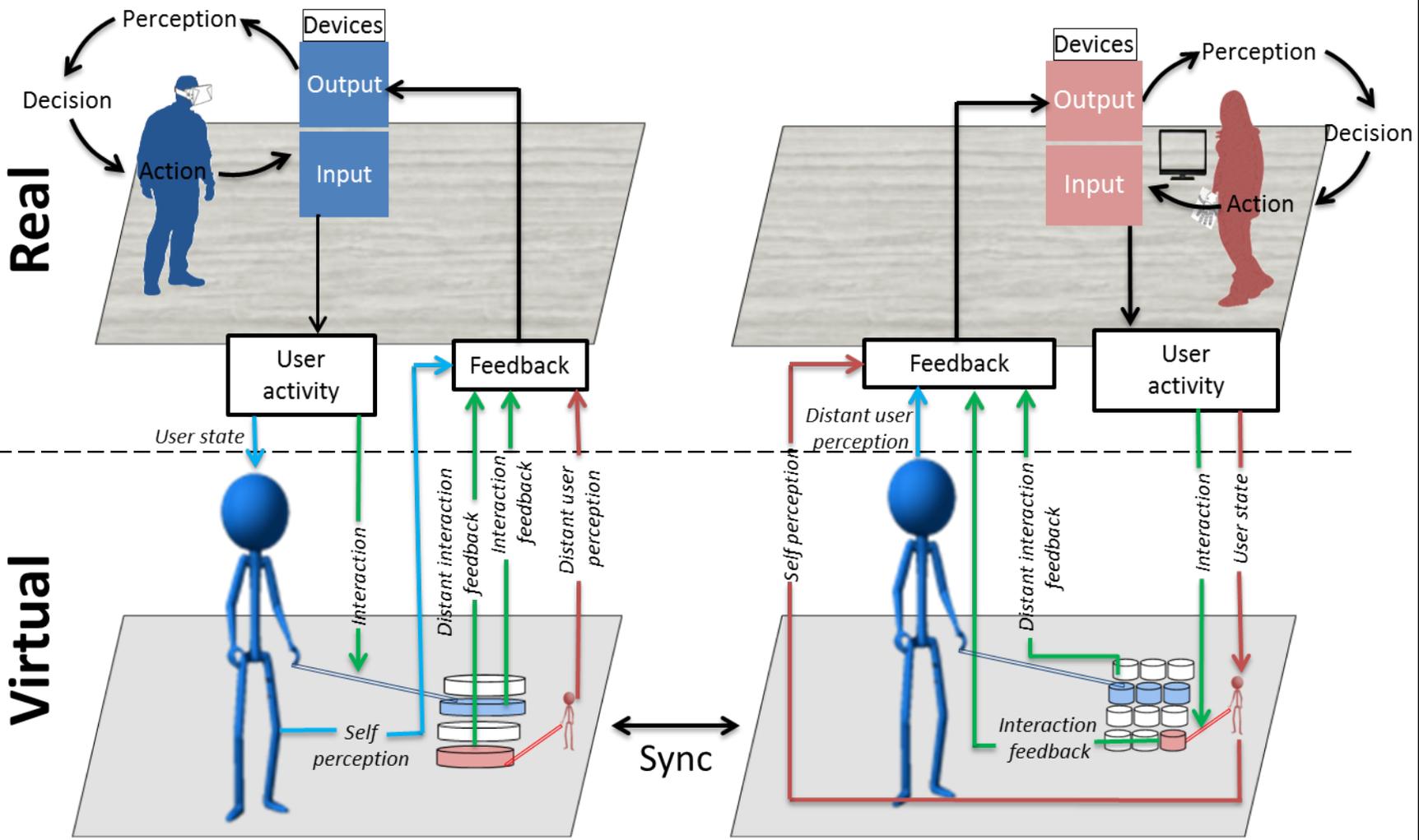


Figure 2.7: Integration of personal virtual spaces in our awareness model, partially synchronized between remote users. This example represents a multi-scale CVE. The blue user has a global PoV on the shared CVE, while the red user has an immersive PoV. Displayed content is slightly different for both users (the blue user can only see the big picture of the shared content aggregated into clusters of smaller parts that can only be seen by the red user; the cluster selected by the blue user makes all the parts of this cluster turn to blue in the red user's virtual personal space; the selection of a part by the red user makes the entire cluster turns to red in the blue user's virtual personal space), while the personal spaces are still partially synchronized in order to maintain consistency in the shared CVE to maximize awareness capabilities of each collaborator.

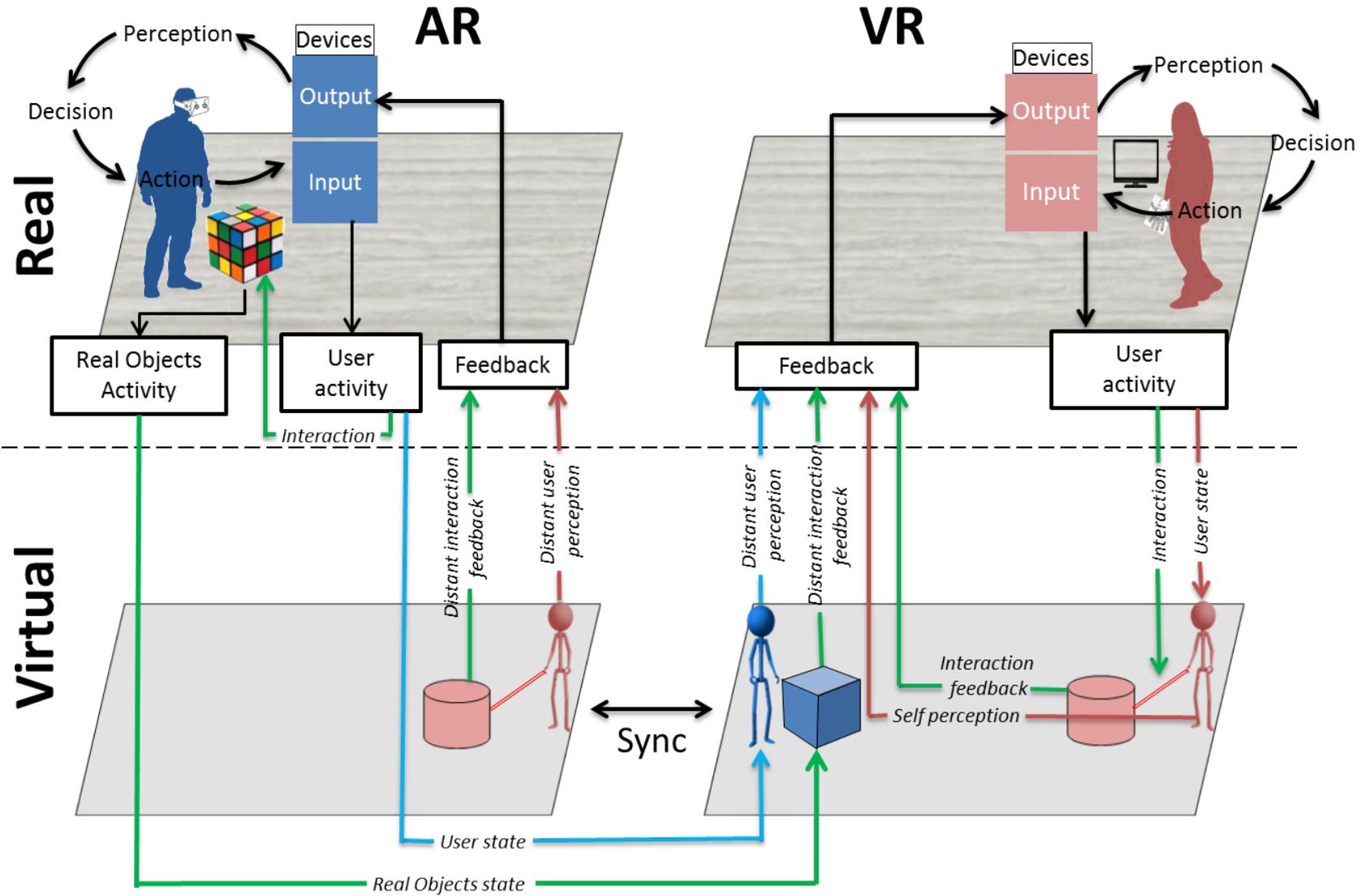


Figure 2.8: Our model enables to describe asymmetrical MR collaboration. The blue user has an AR setup and the red user a VR setup. The real object in the real AR user space is represented as the blue cube in the virtual VR user space. The red cylinder is a shared virtual object represented in both virtual spaces. We can notice that the AR user representation is only presented in the virtual VR user space; its counterpart for the AR user is his/her direct perception of his/her own body in his/her real space.

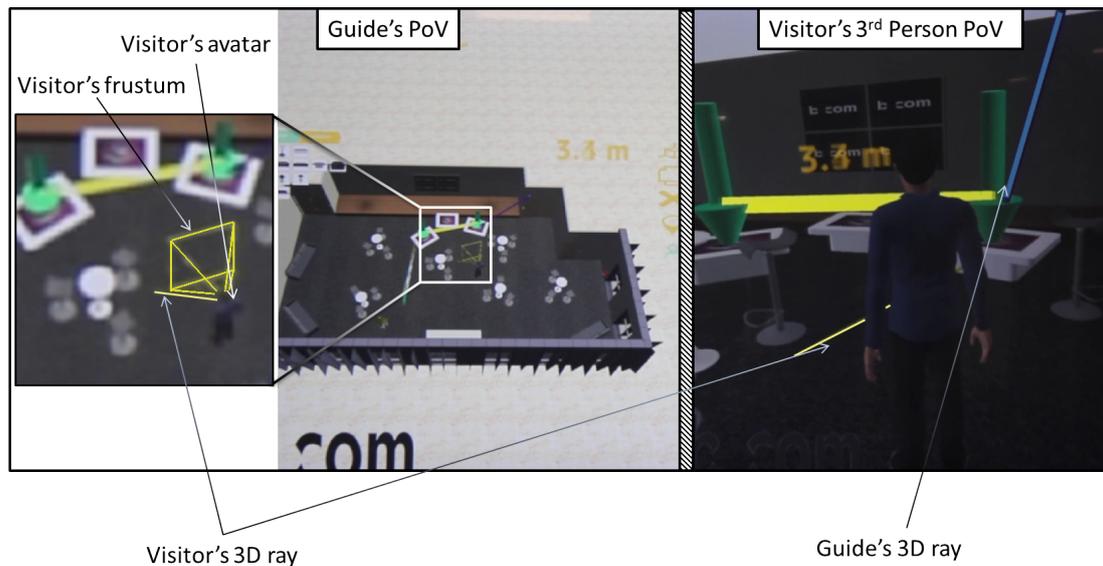


Figure 2.9: Illustration of an asymmetric collaboration with two complementary PoVs [Le Chenechal et al., 2015]. The guide has a global PoV of the whole room (left), while the visitor is immersed in it at 1:1 scale (right). Here, the illustrated visitor's PoV is a 3rd person PoV slightly behind the 1st person PoV actually perceived by the visitor user. On the left guide's zoomed viewport, we see that the visitor's avatar is pretty tiny, and thus it can be tricky to be aware of his/her focus. The frustum can help to understand his/her perceptual capability. We can notice that the interaction tools of both users are represented for themselves and for the others. On the right, the guide's 3D ray is like '*coming from the sky*', which can be complicated to apprehend for novice users.

2.3 Users Representation

Users representation in MR is essential to firstly enable presence within the VE and secondly to add remote activities perception and co-presence, and thus to maintain awareness [Noh et al., 2015]. For one's self presence within the VE, representation of the full body is not necessarily needed because we can not see our own head for instance (except as a reflection on a reflective surface). However, having a virtual body in First-Person Viewpoint (FPV) allows to look downward and, for instance, to see one's chest and legs, which reinforces the sense of embodiment (and thus the sense of presence) due to a matching between proprioception and visual feedback of one's own virtual body [Kilteni et al., 2012]. Even more important, MR aims to provide natural interactions in the VE, thus the use of the hands to act on virtual content is the most natural way to provide such interactions [Mizutori et al., 2012]. Interaction techniques create a link between interaction tools and hands motion. These interaction tools are the virtual items allowing the user to interact with virtual object using his/her hands motion as input. Thus, the representation of these interactive tools in the VE is also very important in order to be able to understand his/her interactions in real-time and to be able to predict the consequences of his/her actions within the VE. As it is linked to hands motion, the explicit representation of the hands, and even of the whole arms, depending of the applicative context, is also useful in order to enhance the embodiment feeling as well as to improve presence and interaction performances.

When dealing with multi-users VE, it becomes even more important to represent users

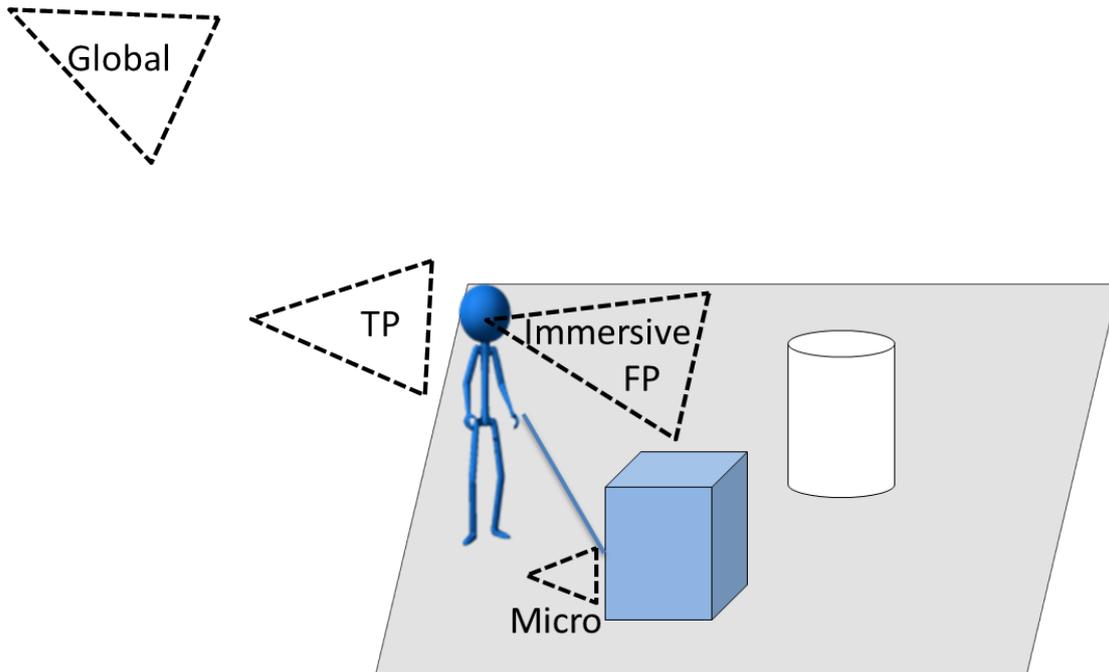


Figure 2.10: Illustration of different PoVs available in a multi-scale VE. PoV *FP* is the egocentric, also called immersive, 1st person 1:1 scale PoV. Another user, with *TP* PoV, can stay behind this 1st person PoV, it is called 3rd person PoV. Now, considering different scales, a user can have a *global* PoV on the whole scene, or at the contrary have an accurate PoV, called *micro* PoV, on a specific area of the VE thanks to its tiny scale within the VE.

within the VE. In addition to be used as a way to feel present in the VE, this representation allows collaborators to be aware of others' presence as well as activities through others' interaction tools representation. In particular, when performing a shared task on a same object, it allows to understand the consequences of combined actions from multiple users who can synchronously act on a shared virtual content in a collaborative, and possibly concurrent, way. Moreover, the collaboration can become even more efficient if collaborators are able to understand the others' field of perception in an implicit way. It allows more natural communication based on visual feedback without a verbal communication overload that can decrease the collaborative interaction performances. For instance, being able to understand the limits of the others' FoV greatly improves coordination because collaborators can know at each time what the others are able to see or not. The use of an avatar with head motion, or even gaze tracking [Steptoe et al., 2008] can handle this feature. Other approaches, as the use of a virtual frustum that represents the actual geometry of collaborator's FoV [Fraser et al., 2004] can also be used as visual feedback to provide remote perception capability in a more precise, but less natural way. Figure 2.9 illustrates an example of users' representation and their interactive tools.

2.4 User's Point of View

In a multi-scale VE, users are able to have different viewpoint scales on their shared VE. Thus, they can benefit from different and complementary PoVs on the same synchronized content. Figure 2.10 explains the terminology given to each kind of PoV. In this collaborative con-

text, it can improve the task efficiency in terms of completion or accuracy. For instance, in the domain of scientific visualization, some global schemes can only be observed at huge scale, e.g. when viewing the content globally, while details need a tiny viewpoint scale to be revealed. For this kind of asymmetric collaboration, users must be aware at all time of the others' activities and capabilities according to their PoV. As an example, let us consider a two-users collaboration with asymmetric PoVs, in which one user has a global PoV, called the guide, while the other user, called the visitor, has a 1:1 scale PoV in the content (cf. Fig. 2.9). In this setting, the guide's PoV allows him/her to have a global view of the room in order to help the visitor in laying out the furnitures. Moreover, the visitor has the ability to walk in the room to have a better understanding of spaces at 1:1 scale. Both users can interact on the content, thus, the system must let collaborators be aware of others actions and perception capabilities despite they do not progress at the same scale. In particular, to be able to efficiently help the visitor, the guide must be aware of what the visitor is looking at. A tricky part in this is to evaluate at which moment the visitor perceives a pointed item from the guide's ray '*coming from the sky*'. Indeed, even if verbal communication can help, it is not necessarily the best way to convey this kind of information, especially in pointing task with many objects with similar appearances. Thus, we can try to use additional visual feedbacks specifically designed for this purpose in order to not add this verbal communication overload that can be inefficient in some situations, especially in asymmetrical PoV settings.

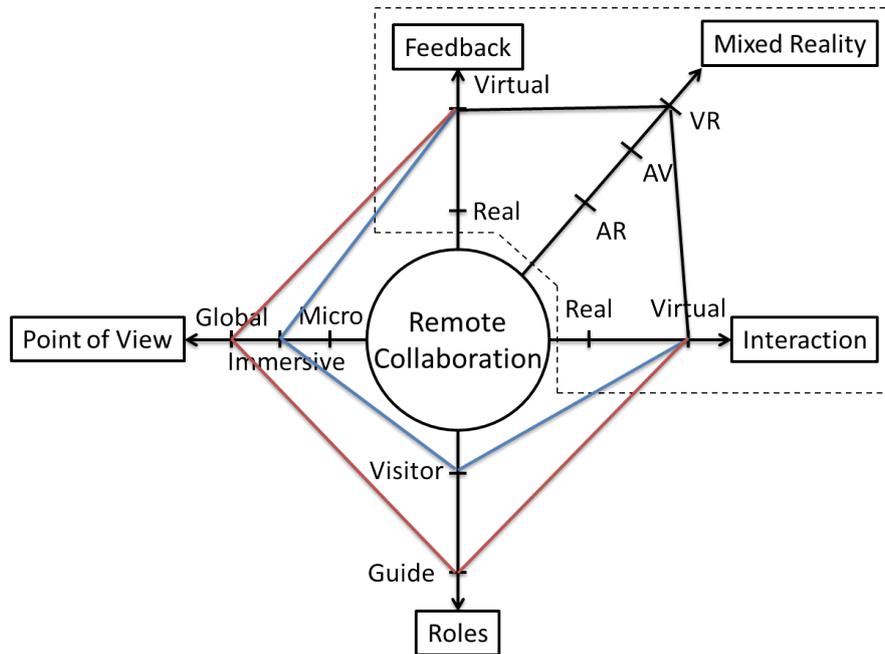


Figure 2.11: The asymmetrical dimensions introduced in our preliminary evaluation about awareness in CVE. Red lines represent the guide’s setting, whereas the blue lines represent the visitor’s setting. Black lines are merged settings between both users.

2.5 Preliminary user evaluation

From the two previous sections, we pointed important awareness issues found in asymmetric collaboration. Especially in a multi-scale VE, when a guide with a global PoV on the scene must help an immersed visitor user, the representation of the visitor could be used to convey additional informations in order to improve the guide’s awareness about the interpretation of his own guiding indications. For instance, from the guide’s PoV, when pointing an object, when does he/she know that the visitor saw it? When guiding the visitor to walk in a specific direction, when does the guide know the visitor understood his/her guidance? Maybe when the visitor begins to walk in the desired direction? Or maybe the visitor did not notice and just walks this way without being aware of the guide’s intention? It is this kind of awareness limitations that we aim to explore in a early preliminary user study. Moreover, other use cases can suffer from these kind of awareness issues such as remote learning if a teacher explains a task or shows something that must be perceived and analyzed by a learner; project review or tele-maintenance are also cases that need to provide a proper understanding of what others have seen and analyzed from our own indicative interactions. Figure 2.11 illustrates the asymmetrical dimensions we introduced in this study in order to explore roles and PoV asymmetry in remote collaboration.

2.5.1 Use case overview: Asymmetric VR remote guiding

We consider a system that handles two distant users in an asymmetric way (note that a symmetric setup could also be relevant). The first one, named the guide, has a global view of the scene thanks to a 3/4 top view of the whole scene (or at least a large part of it). The second one, the visitor, is immersed in the VE with a first-person view of the scene. Figure 2.12 illustrates

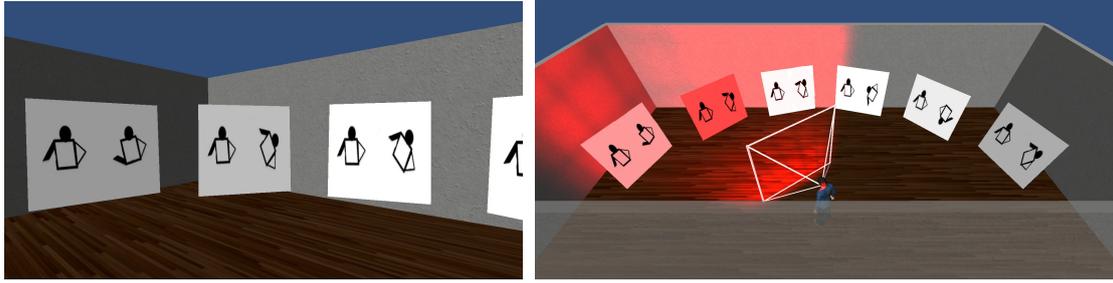


Figure 2.12: (left) The visitor's immersive view; (right) The guide's global view.

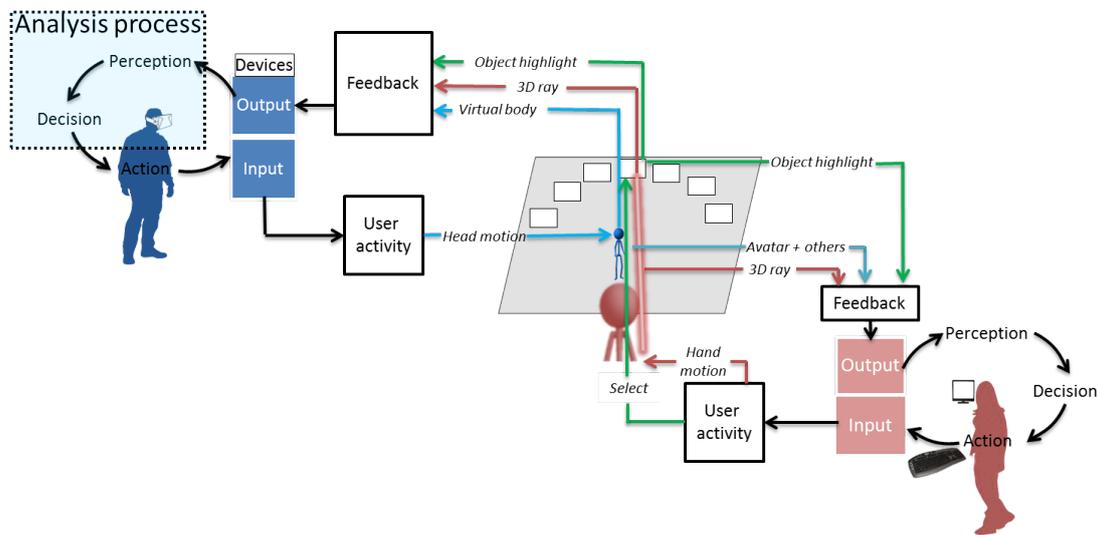


Figure 2.13: Illustration of the experimental use case implementing our CVE model.

these two views.

The task considered in this CVE is a very simple and common one: the guide points to an object (with a 3D ray, for instance) that the visitor must find, perceive and analyze. Most CVEs provide this kind of collaborative feature which allows visual communication. However, classical awareness techniques are symmetrical. This means that if the pointer highlights the object, it will be highlighted in the same way and at the same time for both users. This technique gives no information to the guide that could help him to see that the visitor has perceived and analyzed what he pointed to. The only way to get any idea of this is to estimate what the visitor can perceive thanks to his view frustum and his avatar, for instance, and to estimate the accurate duration needed to analyze the meaning of what has been pointed to. This analysis process can take more or less time depending on the complexity of the object, especially if this part of the scene pointed by the guide requires a reasoning of the visitor to send back a comment or an answer. Thus, the duration of this process must be accurately estimated by the guide, a hard task to accomplish, especially if he has no idea of the experience of the visitor concerning the task. Figure 2.13 implements our model for this specific use case and highlights the analysis process achieved by the visitor to interpret guiding informations conveyed by the guide's 3D ray.

2.5.2 Experimental apparatus

We note that none of the existing solutions takes into account the awareness of what others currently analyzed regarding a remote informative interaction, such as an object pointing that requires a reasoning to be understood. We think this would be an interesting feature to provide in order to improve collaboration, especially in a non-verbal situation [Dourish and Bellotti, 1992]. Indeed, in some cases, verbal communication cannot be used, such as when users speak different languages, when environments are too noisy or when a user is deaf-and-mute. Thus, in this study, we focus on this particular context **without any available audio channel**.

The visitor used an HMD to be immersed in a room containing six boards from among 66 possibilities. These boards presented pictures. Each picture was composed of two figures: the left one was always the same image of reference, whereas the right one could be the rotated image of reference or a different one. The task of the visitor consisted in finding the board pointed to by the guide, and then to answer the question: Are these two figures identical modulo a 2D rotation? This task is called “2D mental rotation” in the psychology literature [Cooper, 1975]. This literature proves that everyone can achieve it with few performance differences between people. Thus, this 2D mental rotation task was a reasonably simple task that nevertheless required an analysis process that simulated the one described in our new asymmetric loop.

The guide’s task (e.g the subject’s task) consisted in estimating the moment when the visitor had analyzed the board and answered. With this task, we wanted to evaluate the estimation performances of the guide regarding the visitor’s activity and the analysis process achievement.

2.5.2.1 Independent variables

Estimation features. In our experiment, the visitor was always presented as an avatar with a frustum representing his/her exact field of view ($\neg\otimes$). Feature \otimes added a squared spotlight matching the frustum dimensions and highlighting the objects in the visitor’s field of view (see Figure 2.12).

Safe-explicit feature. It implements an acknowledgment from the visitor to the guide. In our experiment, an arrow (\Downarrow) was presented to the guide as soon as the visitor thinks that he/she has solved the mental rotation by pushing one of the two physical buttons to answer the binary question. This notification is directly sent to the guide and is displayed in the VE through an arrow. (cf. Figure 2.14). We assumed that, for the visitor, the time between the decision following the analysis process and the answer recorded by the system was insignificant (less than 100ms, as observed in [Card et al., 1986]).

Single and successive tasks. A single task did not cause a perceptible behavior of the distant user when the visitor completed his/her analysis, while a successive task brought about implicit feedback due to a specific behavior of the distant user when the visitor finished his/her analysis. In our experimental process, the most obvious implicit indication was the motion of the visitor who is looking for a new task as soon as the previous one is accomplished. For example, if the visitor looked at the board indicated, then looked elsewhere, it could suggest that the visitor had completed the analysis of the current task. To simulate this behavior, we gave instructions to the recorded visitor. In one case, called **single task** (\dagger), we asked the visitor to keep looking at the board after the completion of the task until the next iteration. In another case, called **successive task** (\ddagger), we asked the visitor to return directly to a phase of VE exploration after solving the task (\ddagger).

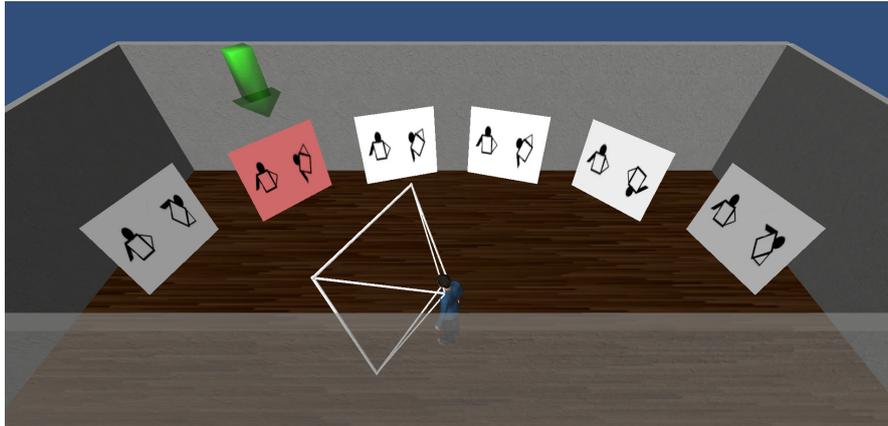


Figure 2.14: Illustration of our implementation of the safe-explicit feedback \Downarrow for this experiment from the guide's point of view.

Combining the three variables of the experiments led to eight conditions. Every subject met all the conditions.

2.5.2.2 Experimental protocol

We designed the experiment in order to be able to evaluate the guide's perception of the distant visitor's activity. Thus, if we had used a real human visitor for each trial, it could have skewed our experimental results due to heterogeneous behaviors. To solve this issue, we recorded one real human visitor achieving the tasks in a pre-process phase. Therefore we were able to replay human behaviors in a counterbalancing design for each subject of the experiment without any bias regarding the visitor's behavior.

Afterwards, each participant was first informed about the complete study proceedings. Then, they completed an identification questionnaire allowing us to collect general information about their experience with VR. Next, a demonstration introduced the experiment and gave instructions about the objective for 10min. The experimental manipulation was composed of 56 iterations (8 conditions * 7 iterations) and lasted 10min.

Each condition iterated on seven boards from among the 66 available ones (using a counterbalancing order). We ensured each board was pointed to at least once for each condition. One iteration was decomposed as follows:

1. the system simulated a distant pointing of the guide: the symmetric awareness technique effects board flicking;
2. the system started to replay the recorded visitor's behavior sequence;
3. the visitor searched for the board indicated;
4. the visitor found it and watched it;
5. the visitor solved the task by analyzing the mental rotation;
6. only in \Downarrow conditions, a safe-explicit feedback was send to the guide meaning that the visitor had completed the analysis of the board pointed to.

Moreover, we ensured that the pre-recorded visitor always completed the task before a 10s timeout (actually he usually completed it in under 5s). During an iteration, the subject's task

consisted in clicking when estimating that the visitor had completed the board analysis. After each validation, we asked the subject the chosen strategy for estimating when the visitor had completed the analysis of the indicated board. The subject could choose between five options:

- *no specific strategy*
- *self mental rotation*: estimation based on the subject’s own performances in achieving the mental rotation task;
- *time count*: estimation based on a roughly evaluation of the time needed to accomplish the mental rotation task;
- *visitor motion*: estimation based on the behavior of the visitor perceived through the avatar and frustum (and the spotlight in \otimes condition);
- *other: type any strategy*: with this text field, subjects could specify any strategy they used.

We did not explicitly propose *the apparition of the arrow* (\Downarrow) as strategy, in order not to influence the subjects’ answer. But it could be specified in the *other* text field and subjects was informed that they could use copy/paste keyboard shortcuts to repeat an already used strategy without tapping it each time.

We briefly explained the meaning of each proposal in the trial phase. For each condition change, a black screen displayed the state of the three independent variables for the next conditions. Three icons always remained visible to the subjects to remind them of the current state of independent variables.

At the end, participants were asked to fill in a questionnaire to provide their impressions, comments and subjective judgments about the estimation features, the safe-explicit feature and the tasks with:

- a dichotomous preference test between the estimation features ($\neg\otimes / \otimes$), the safe-explicit feature ($\Downarrow / \neg\Downarrow$) and the tasks (\dagger / \ddagger);
- a Likert-scale (1: Not certain at all, 4: Totally certain) with the following questions about the estimation features, the safe-explicit feature and the tasks:
 - “Are you confident that the visitor has seen the board indicated?”
 - “Are you confident that the visitor has completed the analysis of the board?”

2.5.2.3 Measurements and hypotheses

Measurements We measured the time of completion of the task by the visitor and its estimation by the guide. Thus, we compared the estimation accuracy of the analysis process of the visitor by the guide according to each condition. The goal was to improve this estimation and thus the narrower the gap between collaborators’ validation, the better the guide’s estimation.

It should be noted that, in order to remove possible bias and to homogenize collected results by removing uncontrolled variables, we analyzed only data that respected the following conditions: the guide (e.g the subject) validated before the 10s timeout and the response to the mental rotation task was ‘yes’.²

Hypotheses We assumed some hypotheses:

- H1: In successive task (\ddagger), the additional estimation feature (\otimes) improves the estimation accuracy because it enhances the representation of the visitor’s perception capability.
- H2: In single task (\dagger), the safe-explicit feedback (\Downarrow) improves the estimation accuracy because estimation features do not provide enough feedback about the visitor’s activity.

2. Some boards ($\sim 10\%$) still presented different figures in order to keep the recorded visitor focused on his task.

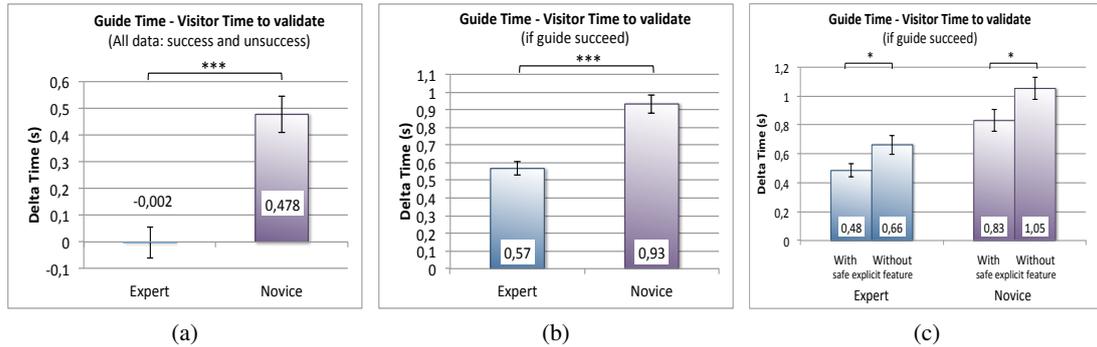


Figure 2.15: Difference of time between the visitor validation and the subject estimation: (a) globally, (b) when subject validated after the visitor, (c) and regarding the availability of \square .

2.5.3 Results

2.5.3.1 Participants

Our panel was composed of 20 subjects aged from 23 to 54 ($M = 32, SD = 8$). There were 12 males and 8 females. Eight of them were considered as experts and 12 as novices. This consideration was based on their personal experience in VR and, more broadly, in 3D and/or collaborative video games. We considered a subject as expert if he had spent more than an hour per week during the last weeks, and already spent more than 14 hours per week during their lifetime using these kinds of applications. Our subjects had various backgrounds: PhD students, R&D engineers, communication and human resources staff, managers and assistants.

2.5.3.2 Analysis estimation

We analyzed the delta time between both validations (visitor and guide). If this difference is positive, that means that the guide validated after the visitor. If the difference is negative, that means that the guide validated before the visitor.

Figure 2.15a illustrates the significant difference between novices and experts ($F(1, 939) = 19.78, p < .001$). Novices ($M = 0.48, SD = 0.07$) validated slower than experts ($M = 0.00, SD = 0.06$). Experts validated at the same time than the visitor, and even before the visitor. This was a main result: some experts validated before the visitor. Subjects had to validate when they were convinced that the visitor had analyzed the board. When the participants validated before the visitor, we considered that they failed.

In the case of safe-explicit feature (\square), an arrow appeared when the visitor just completed his current task. The mode (\dagger or \ddagger , \square or $\neg\square$, \otimes or $\neg\otimes$) was presented at the screen. So, the subject knew if an arrow would appear or not. Rationally, in the case of safe-explicit feature (\square), subjects had just to wait the apparition of the arrow to validate.

We also analyzed the cases of subjects' success, e.g when the guide took the mental decision after the visitor. As action's duration to click was assumed to be less than 100ms, as observed in [Card et al., 1986], we assumed the mental decision was the time to click minus 100ms.

Figure 2.15b illustrates the significant difference between novices and experts ($F(1, 561) = 15.76, p < .001$). Novices ($M = 0.93, SD = 0.005$) validated slower than experts ($M = 0.57, SD = 0.04$). Results show that experts are better than novices to evaluate the right time the visitor achieved the mental rotation analysis.

The analysis of the additional estimation feature results (\otimes vs $\neg\otimes$) gave no significant differences. The spotlight was not a useful additional estimation feature in this study. **As the results were not significant, we do not analyze the effect of this feature (\otimes) in the following.** Moreover, H1 is not validated and we will discuss this result in section 2.5.4.

On the contrary, the safe-explicit feature results (\sqcup vs $\neg\sqcup$) gave significant differences for experts ($F(1, 148) = 5.34, p < .05$) and for novices ($F(1, 411) = 4.304, p < .05$). Figure 2.15c shows that experts and novices validated slower without (expert: $M = 0.66, SD = 0.07$; novices: $M = 1.05, SD = 0.07$) than with the safe-explicit feature (experts: $M = 0.48, SD = 0.04$; novices: $M = 0.83, SD = 0.08$) The safe-explicit feature that implemented an acknowledgment from the visitor to the guide improves the estimation accuracy, as stated in H2 for single task (\dagger).

In our experiment, the safe-explicit feature was represented by an arrow above the board for the guide indicating that the visitor had just completed his task (cf. Figure 2.14). Even when the participants had to validate as soon as they saw the arrow, the experts were better. This result could be explained by the skills of experts in VE.

In the single task (\dagger), the analysis of the time to validate gave not any significant difference between experts and novices, even if experts ($M = 0.7, SD = 0.09$) were quicker than novices ($M = 0.8, SD = 0.07$). When the visitor were motionless after the completion of his task (\dagger), experts and novices met the same difficulties to evaluate the right time to validate. Obviously, in this case of uncertainty, the safe-explicit feature helped the subjects to decide ($F(1, 250) = 4.6, p < .05$). We could appreciate how the safe-explicit feature guided the decision-making process in two ways:

- subjects validated quicker with (0,67) than without (0,91) the safe-explicit feature;
- the standard deviation was smaller with the safe-explicit feature ($SD = 0.07$) than without it ($SD = 0.09$).

The safe-explicit feature was truly useful when the visitor’s behavior did not clarify when the analysis process was ended.

In the successive task (\ddagger), the analysis of the time to validate gave significant difference between experts and novices ($F(1, 327) = 6.6, p < .05$). Novices ($M = 0.84, SD = 0.06$) validated after experts ($M = 0.58, SD = 0.04$). In this case, the safe-explicit feature significantly helped experts, not novices. Indeed, experts validated earlier with ($M = 0.39, SD = 0.01$)

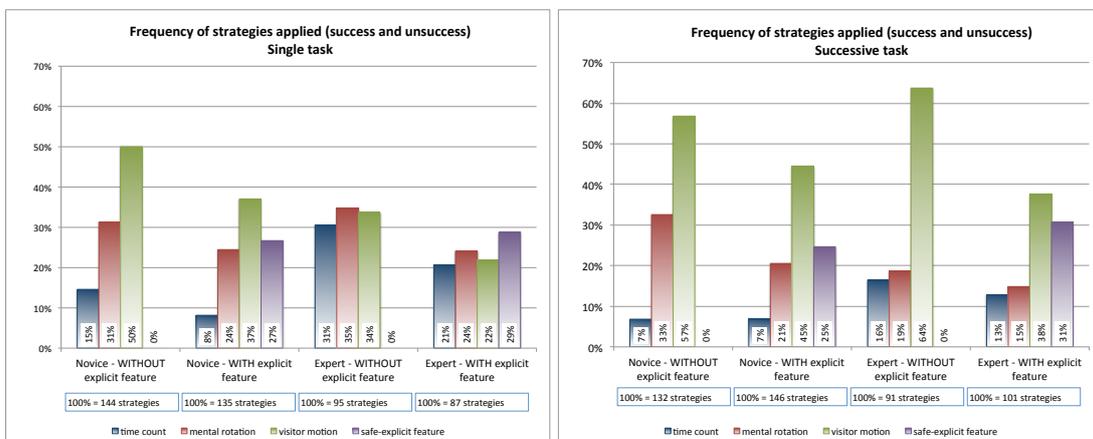


Figure 2.16: Strategies applied according to the task and the expertise.

than without the safe-explicit feature ($M = 0.82$, $SD = 0.03$). The difference was significant: $F(1, 19) = 152.7$, $p < .001$. Even if novices validated earlier with ($M = 0.82$, $SD = 0.11$) than without the safe-explicit feature ($M = 0.87$, $SD = 0.06$), the difference was not significant. This result could be explained by the strategies applied (cf. subsection 2.5.3.3)

The visitor's behavior was an implicit feedback; as the experts validated earlier, experts were better than novices to understand this feedback.

2.5.3.3 Estimation strategies

The strategies applied depended on the tasks (\dagger / \ddagger), the situations ($\square / \neg\square$) and the groups (novices/experts) (cf. Figure 2.16). In the single task, for each situation ($\square / \neg\square$), novices applied the visitor motion strategy more often than experts what has no real meaning as motion does not express the task achievement in this condition. For each situation, experts applied time count and mental rotation strategies more often than novices. In the successive task, the most common strategy applied was the visitor motion strategy. Without safe-explicit feature, experts applied the visitor motion strategy and the time count strategy more often than novices. And novices applied self mental rotation strategy more often than experts.

Figure 2.17 illustrates the success of strategies for validation after the visitor validation. For each strategy, we calculated the ratio of times the strategy was applied with success. With a ratio less than 50%, the guide validated more often before than after the visitor. When the subjects validated before the visitor, we considered that they applied an unsuccessful strategy to estimate the right time to validate.

Figure 2.17 shows that the less successful strategies were the mental rotation (47%) and time count (53%) strategies and the most successful strategies were visitor motion (62%) and safe-explicit feature (100%) strategies.

In the mental rotation strategy, the guide realized a mental rotation, with the implicit assumption that the visitor took the same time to analyze the board. In the time count strategy, the evaluation was quite more approximated: the guide did not truly analyzed the board but evaluated the complexity of the rotation and adjusted the time that the visitor took to complete it. These two strategies were based on how the subject analyzed himself the board, not on the visitor's behavior or on the safe-explicit feature. These two strategies illustrated an ego-centric perspective, which was frequently unsuccessful in this study. One of the reasons that explains this result could be the difference of viewpoint between the visitor and the guide (cf. Figure 2.12). The guide had a global viewpoint that allowed him/her to watch the 7 boards. The visitor had an immersive viewpoint with a narrow visual angle that could contain only 3 boards

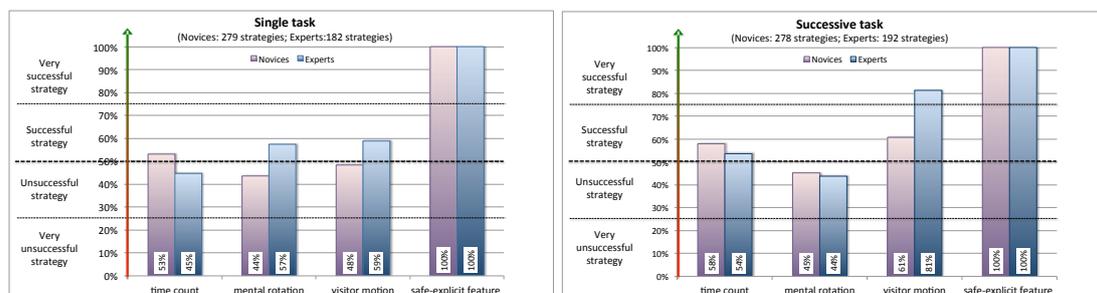


Figure 2.17: Successful applied strategies according to the task and the expertise.

Task	Explicit feature	Type	Novices	Experts
			Strategies applied	Strategies applied
Single task	Without	Most frequent	Visitor motion	Mental rotation
		Most successful	Visitor motion	Visitor motion
	With	Most frequent	Visitor motion	Safe-explicit feature
		Most successful	Safe-explicit feature	Safe-explicit feature
Successive task	Without	Most frequent	Visitor motion	Visitor motion
		Most successful	Time count	Visitor motion
			Visitor motion	
	With	Most frequent	Visitor motion	Visitor motion
		Most successful	Safe-explicit feature	Safe-explicit feature
			Safe-explicit feature	Safe-explicit feature

Table 2.1: The most applied and most successful strategies in all the conditions for experts and novices.

at the time. The guide could see the board pointed earlier than the visitor, thus the guide could validate before the visitor by applying egocentric strategies.

Visitor motion and safe-explicit feature strategies illustrated an allocentric perspective. The guide was waiting for a visitor’s information: a punctual successive behavior or a safe-explicit feedback. The visitor motion strategy was often successful in the successive task, but not so much in single task. Indeed, in the successive task, the visitor looked at the board to analyze it, answered and then moved to look for the next task. Thus, the visitor’s behavior was useful to detect when the analysis was completed. One would have thought that the success score would be higher. The participants told they met difficulties to detect the fixation when the analysis process duration was too short. In this condition, the visitor explored the VE to look for the pointed board, found the board, analyzed it, answered and returned exploring the VE. When the analysis process was short, the guide could perceive the visitor’s behavior as continuous. In the single task, the visitor watched the board to analyze it and stayed on it after answering. So, the visitor’s behavior was not useful to detect when the analysis was completed. This behavior was just useful to detect when the fixation began. Then the guide had to evaluate the time to validate after this fixation. Often they validated too early. Figure 2.17 illustrates this difficulty. The safe-explicit feature was a very successful strategy. Obviously, a validation after the arrow indication was always successful.

2.5.3.4 ROI of strategies applied

We calculated a ROI (Return On Investment) by multiplying the frequency F of the strategy i applied ($0 \leq F_i \leq 1$) by the score of success of this strategy S ($0 \leq S_i \leq 1$) and by adding these results obtained for the four main strategies ($0 < i \leq 4$). $ROI = \sum_i (F_i * S_i)$ with $\sum_i F_i \leq 1$ and $0 \leq S_i \leq 1$. Thus, $0 \leq ROI \leq 1$. The better was the ROI, the better was the global success of the strategies chosen by each group of participants. Table 2.1 summarizes the most applied and successful strategies, and table 2.2 gives the associated ROIs scores.

The analysis of the ROIs proved that experts applied strategies that fit better than novices. This analysis proved that the highest ROIs in successive task were due to the visitor motion

Task	With / Without explicit feature	Novices	Experts
Single task	Without feature	0,47	0,51
	With feature	0,59	0,68
Successive task	Without feature	0,61	0,73
	With feature	0,58	0,71

Table 2.2: ROI of the strategies applied illustrating their global success.

and to the safe-explicit feature strategies. The visitor's behavior was an implicit feedback that worked. We proved that as the experts validated earlier, they were better than novices to use this feedback. The safe-explicit feature was truly useful when the visitor's behavior did not clarify when the analysis had been achieved. Experts used more often than novices this feature to validate.

2.5.3.5 Questionnaire results

Question: "Are you confident that the visitor has seen the board indicated?" Globally, all the participants were very confident that the visitor had seen the board indicated (> 82%) (just under for novices in the successive task: 67%). The additional estimation feature with the spotlight was more appreciated, particularly by novices. The participants preferred the single task to be firmly convinced that the visitor had seen the board. To optimize confidence that the visitor has seen the board, the best configuration was: spotlight (⊗) and single task (‡).

Question: "Are you confident that the visitor has completed the analysis of the board?" Obviously, with the safe-explicit feature, the participants thought with a higher degree of confidence that the visitor had completed the analysis of the board. The participants preferred successive task to be firmly convinced the visitor had completed his analysis. "If the visitor moves after steadying, that means he completed the task, so I validate" said numerous participants. To optimize confidence that the visitor had achieved his analysis, the best configuration was: safe-explicit feature (⊥) and successive task (‡).

2.5.4 Experimental design comments

The proposed additional estimation feature, implemented as a spotlight matching the frustum dimensions, did not prove its usefulness (e.g we did not validate H1). An explanation could be that the boards were distributed on a horizontal half-circle around the visitor. Thus, the rotation of his head was almost around only one axis. This partially decreases the 3D aspect of the task as the guide can easily estimate thanks to the viewing frustum which objects are visible by the visitor (no depth perception ambiguity). To solve this issue, we could use a sphere to dispose the boards. The head would rotate around the three axis, and it would be more difficult to estimate his field of view without the spotlight. This would be a more realistic use case for CVE, in addition to provide more interactivity, and would also be interesting to generalize the

new asynchronous loop we propose to enhance awareness in CVE. Thus, it could extend this work in more complex scenarios while improving limits of this experiment.

2.5.5 Discussion

We ran a user study in a CVE involving a simple cognitive task called 2D mental rotation [Cooper, 1975]. We analyzed the strategies collected from the participants to estimate a time when a pre-recorded collaborator, called the visitor, achieved his analysis of a pointed board showing a problem to solve. Self mental rotation and time count strategies illustrated an egocentric perspective. They were frequently unsuccessful in this study, e.g by applying these strategies, the participants validated before the visitor. One of the reasons to explain this result could be the asymmetric setup in terms of viewpoint. If the guide (e.g the subject), with a global viewpoint, applied egocentric strategies to analyze the pointed board, their validations were earlier than the visitor's validation due to his immersive viewpoint.

In the opposite, visitor motion and safe-explicit feature strategies illustrated an allocentric perspective. The guide better took into account the visitor's activity, which led to an enhanced estimation accuracy. In this study, the guide could use implicit feedback such as the visitor motion to estimate the completion of the visitor's analysis process. Moreover, in some conditions, we provided a safe-explicit feature that enabled the guide to know the exact moment the visitor completed his analysis. This last has proved to be very useful for experts and novices that used it successfully in most cases.

We also differentiated two groups of users. Experts and novices applied strategies to estimate the time to validate:

- taking into account the visitor motion;
- waiting for a safe-explicit feedback;
- achieving the same analysis as the visitor had to achieve (2D mental rotation);
- counting mentally a time (few seconds) depending on the analysis complexity perceived by the guide.

But, experts and novices did not succeed in the same manner, because experts chose their strategies in a better way than novices. Experts were able to select their strategies in function of the condition better than novices. For example, in single task and with an available safe-explicit feature, novices applied visitor motion strategy whereas experts applied the most successful strategy: the safe-explicit feature.

To sum up, experts and novices did not apply the same strategies to interpret awareness features of the distant activity. Even if we provide adequate awareness features, their interpretation stay user-dependent (especially according to the expertise). Thus, future work should take into account the users' expertise to enhance awareness and improve performances by adapting the interface.

2.5.6 Awareness proposal: an additional asymmetric loop

In this exploratory user study, we focus on the awareness of collaborator's activities. It taught us that users interpret feedbacks in different ways in order to build their own awareness of a shared VE. Especially, experts are able to adapt to available awareness features and react to the presence of explicit acknowledgment when available. Thus it could be a useful feedback to provide to the expert in remote guiding context. Therefore, we propose to add an asynchronous awareness loop in an asymmetric CVE (in terms of roles and viewpoints) in order to propose a new way to be aware of what the others currently analyzed in the CVE.

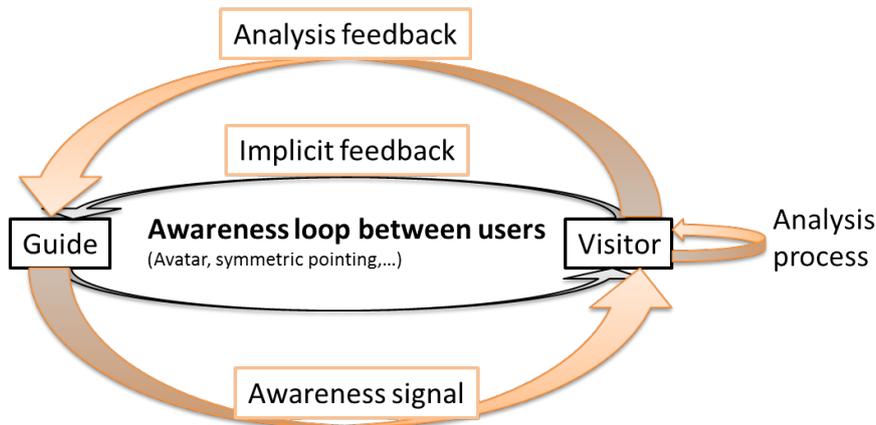


Figure 2.18: The internal black loop shows the classical continuous awareness loop and the external orange loop illustrates the new asymmetric loop that handles the analysis feedback.

To this end, we propose a new concept for collaborative awareness. Indeed, in a collocated (where users share the same physical and virtual environment) or a remote setup, users can only interact in an asynchronous way since they are independent [Card et al., 1986]. This asynchrony is not induced by the system design, but because of human-human collaboration. Indeed, human-human collaboration can target perfect synchrony, but can not achieve it in practice because multiple individuals are involved. Even if they co-manipulate a shared object and try to synchronize their motion, they are still two individuals. For example, the main difficulty in extending a bimanual manipulation toward a cooperative manipulation is the lack of the proprioceptive sense of the others [Le Chénéchal et al., 2014a]. Actually, the lack of proprioception induces a delay in awareness of other user's activities that induces an asynchrony. This limitation can explain the meaning of the asynchronous aspect that we introduce (differently from the classical definitions of system-based asynchronous interactions that take place at very different times as in [Gericke et al., 2014]). Indeed, in this sample case, the lack of proprioception generates a desynchronization between collaborators' behaviors, and thus an asynchronous collaborative interaction considering a very high frequency for time-step. We propose to manage this feature with a new asymmetric awareness loop that aims at improving synchronization of collaborators' behaviors. Especially, we can explain more simply its interest in an asymmetric context as illustrated in Figure 2.12. This asymmetry can be useful in many scenarios, such as in a guiding task, in order to benefit of different capabilities.

In this setting, the guide has a global viewpoint of the scene, and the visitor is immersed in the VE. Here, it can be tricky for the guide to be sure that the visitor saw and analyzed an object he pointed to because:

- their viewpoints are different;
- the duration of their analysis is user-dependent

Thus, we extend collaborative awareness with a new feature able to manage *awareness of collaborators' analysis*. Unlike the other ones, this part takes place in an asymmetric interactive loop that can provide awareness features in an asynchronous way, as explained in Figure 2.18. Referring to our previous experiment, estimation features take place in the classical black loop while the safe-explicit feature is an example of an analysis feedback instantiation (its associated awareness signal is the pointing of a board by the guide that the visitor must analyzed). Using this new loop, we aim to reduce the lack of mutual knowledge, especially concerning what

others currently analyzed.

Due to implicit feedbacks of the classical symmetric loop that implements some workspace awareness features [Gutwin and Greenberg, 2002], some users could estimate the distant activity. But, some misunderstanding are still possible. Thus, even if our new asymmetric loop could bring redundant awareness informations regarding already available implicit feedbacks, some users could use it to improve their estimation of the others' activity.

2.6 Conclusion

In this chapter, we proposed a model that is able to describe an asymmetrical collaborative MR system, and especially that is able to detail available feedbacks and their roles in maintaining awareness for the different users. Then, we conducted an exploratory study with two collaborators sharing a multi-scale CVE. This preliminary study shows difficulties that can be encountered when designing a collaborative system with asymmetrical roles and PoVs. Especially, it has a great impact on the desynchronization of collaborators' behaviors; asymmetrical users' representations and PoVs makes more difficult to target perfect synchrony between remote collaborators' behaviors.

In next chapter 3, we apply our awareness model to remote guiding application. In our first proposal, we goes further in MR collaboration with asymmetrical PoVs. The setting is quite similar to the one studied previously, but this time the visitor (called the operator in remote guiding context) is using an AR setup in order to be guided in performing a task in the real world. Then, in order to improve collaborators' behaviors synchronization, we explore a symmetrical setting in terms of PoVs, and particularly we study the efficiency of shared PoVs. Indeed, since roles, devices and level of virtuality asymmetry are involved by the remote guiding context, we leverage the PoV dimension to enhance remote maintenance in MR.

Then, in chapter 4, we explore an even more sensitive task in terms of users' behaviors synchronization that is virtual co-manipulation. Indeed, in order to achieve this task efficiently, collaborators need to target perfect synchronization while co-manipulating the same object at the same time. With this specific task, we address more users collaborating in a shared VE that our previous use case on remote guiding and we demonstrate how our awareness model is able to describe more complex asymmetrical roles in pure VR remote collaboration.

Chapter 3

Remote guiding

The help of a remote expert in performing a maintenance task can be useful in many situations and can save time as well as money. In this context, Augmented Reality (AR) technologies are able to improve remote guiding thanks to the direct overlay of 3D informations onto the real world. On the other part, Virtual Reality (VR) enables the remote expert to virtually share the same place as the physical maintenance one. Classically in a local collaboration, collaborators are face-to-face and watch the same artifact while being able to communicate verbally and using body language such as gaze direction or facial expressions. This communication is often not possible in a remote collaborative maintenance scenario in which the agent uses an AR setup while the remote expert uses a VR setup. Moreover, as we have experimented in the previous chapter, providing users with adapted interactions and awareness features to compensate the lack of essential communication signals is a real challenge for remote MR collaboration. However, our context offers new opportunities to augment collaborative abilities, such as sharing the exact same point of view, which is not possible in the real life. According to the current task of the maintenance procedure, i.e navigation to the correct location or physical manipulation, the remote expert may choose to freely control his/her own viewpoint on the distant workspace, or, on the contrary, may need to share the viewpoint of the agent in order to better understand the current situation of this agent. First, we focus on the navigation task that is essential to complete the diagnostic phase and to begin the maintenance task in the correct location; then, we present a novel interaction paradigm, implemented in an early prototype, that we proposed to let the guide show the manipulation gestures to accomplish in order to achieve the physical task needed to perform the maintenance procedure. These concepts have been evaluated, thus we can provide guidelines for future systems targeting efficient remote collaboration in MR environments.

3.1 Introduction

Mixed Reality (MR) is a promising research area that proposes to mix real world with virtual artifacts. It offers natural ways to display virtual content taking advantage of real world referencing in order to ease interactions. This leads to more immersive and intuitive systems, and thus improves user performances.

On another hand, remote collaboration is a powerful tool in many situations: distant meeting, social communication, entertainment, teaching, etc. Particularly, collaborative maintenance involving a remote expert helping an agent to perform a maintenance procedure has many advantages in terms of time and money savings. However, many challenges come with remote collaboration:

- perceiving the distant place in a comprehensive way;
- being able to communicate using shared referential;
- understanding remote activity and perception.

We distinguish two research fields that work on these topics. First, Computer-Supported Cooperative Work (CSCW) focuses on remote collaboration with desktop computer. It mainly proposes systems handling remote meeting and concurrent document edition while maintaining consistency between the distant physical places [Rama and Bishop, 2006]. Second, research on Collaborative Virtual Environment (CVE) takes advantage of VR technologies to propose remote collaboration in 3D VE with natural perception abilities and more intuitive interactions. These systems mainly provide shared 3D virtual worlds in which remote collaborators can interact on 3D virtual data; and also, immersive telepresence systems that aim to decrease the real distance between users to let them communicate as if they were in the same room [Zillner et al., 2014].

Remote guiding of an agent performing a physical task helped by an expert has already proved its usefulness in many proposed systems [Robert et al., 2013, Huang et al., 2013b]. It can be used to achieve punctual unknown procedures, decreasing the time and the cost of the interventions and allowing a single expert staying in the same place to help several potentially dispersed remote agents. Still, one of the hardest part of these systems is to let collaborators understand what is happening in the distant places. It is an essential feature to be able to collaborate efficiently without misunderstanding. In this work, we propose innovative interaction techniques always focusing on this awareness aspect in remote collaboration situations. Especially, we focus on collaborative remote maintenance scenarios that imply several constraints:

- the agent performs in the real world with 1:1 scale interactions and is guided with an AR interface;
- the remote expert can only help through virtual interfaces and representations.

This leads to imagine different settings for the remote expert interfaces and interactions. First, we can provide a symmetric setting in which the remote expert interacts, as close as possible, as if he was in the agent’s workspace in a 1:1 scale setting. Second, we can also imagine an asymmetric setting providing a kind of World In Miniature (WIM) view [Stoakley et al., 1995] that allows the expert to have a more global view on the shared scene, while the agent still belongs to the real 1:1 world. Both conditions have advantages depending on the current guiding task, and generate different user experiences and performances. In the same way, the agent’s AR interface can be based on a head-worn or a hand-worn display and cameras. Head-Mounted Display (HMD)-based AR can intrinsically handle co-location of the guidance, and thus avoids indirect mapping into the real world. On the contrary, Hand-Worn Display (HWD)-based AR does not directly display information into the real field of view of the agent. However, this setting allows more freedom for the camera motion that is used by the remote expert to perceive the state of the real workspace; but it does not let the operator’s hands free, which adds a constraint to the operator while performing the physical task.

First, section 3.2 discusses about related work, then section 3.3 presents an overview of our proposals. Then, each system is developed respectively in section 3.4 and section 3.5. Last, we conclude this chapter and discuss global results in section 3.6.

3.2 State of the art

In our scenarios, we divided the maintenance activity in two phases. First, the expert must be able to diagnose the remote workspace to find how to perform the maintenance task. Second,

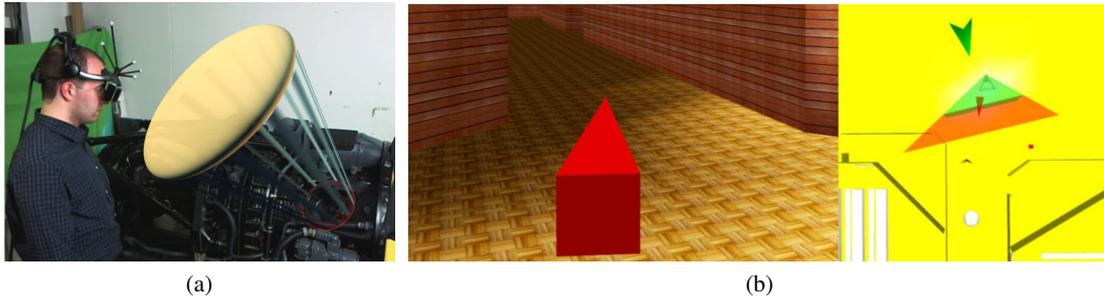


Figure 3.1: Navigation guiding techniques. (a) The para-frustum technique [Sukan et al., 2014] for single user AR. (b) The compass technique [Nguyen et al., 2013] for collaborative application in a multi-scale CVE (The left screenshot is the visitor’s view and the right one is the guide’s view).

the expert must be able to guide the operator in performing the actual physical task needed to achieve the maintenance procedure. Both of these phases require two kinds of interaction for the expert: navigation and manipulation. We can notice that navigation can imply that the expert must be able to remotely navigate within the distant workspace through the shared VE built from sensors moved by the agent, who, of course, is not a robot. In the following, we present the related work concerning collaborative navigation and manipulation guiding.

3.2.1 Navigation

Making a user reach a specific viewpoint is not only used in CVE. Indeed, many single user MR applications need to provide such a feature. Recently, Sukan et al. proposed the para-frustum approach to guide a user in an AR application to reach a set of acceptable viewpoints [Sukan et al., 2014]. This paper presents an advanced technique based on a 3D augmentation that shows to the user the viewpoint to reach (cf. Fig. 3.1a). Compared to classical approaches such as the use of head-up display (HUD), this technique greatly eases the reaching process because of a fully collocated guidance in the real interaction space. However, Sukan et al. highlight the complexity of defining the shape of the 3D augmentation, and thus it seems complicated to extend it for collaborative purposes with an online definition of the shape by a distant user.

In CVE, we distinguish two approaches based on the system design: asynchronous and synchronous guidance. First, asynchronous guidance is often used when a user defines some interesting viewpoint to another user, who can switch from one predefined viewpoint to another. In [Duval et al., 2008], these interesting viewpoints are represented with virtual cameras all around a set of scientific data visualizations. Second, synchronous guidance is the category to which our proposals belong to. It is utilized by a user who interactively defines a viewpoint to reach for another user. Many metaphors have been proposed [Nguyen et al., 2013]: directional arrows, compass (cf. Fig. 3.1b), lit path. However, these techniques work well in an asymmetric setup (i.e. when the guide has a global view of the scene and while the visitor is immersed in it) but are hardly extendable for a symmetric, fully collocated setup. Indeed, even if their implementations can be fully automated (i.e. the guide does not need to define the parameters of the guidance cues himself because the system is able to compute them automatically) they suffer from strict limitations: the loss of naturalness in the guidance process for the guided user because of the interpretation of specific visual cues and the eye strain that can be generated by

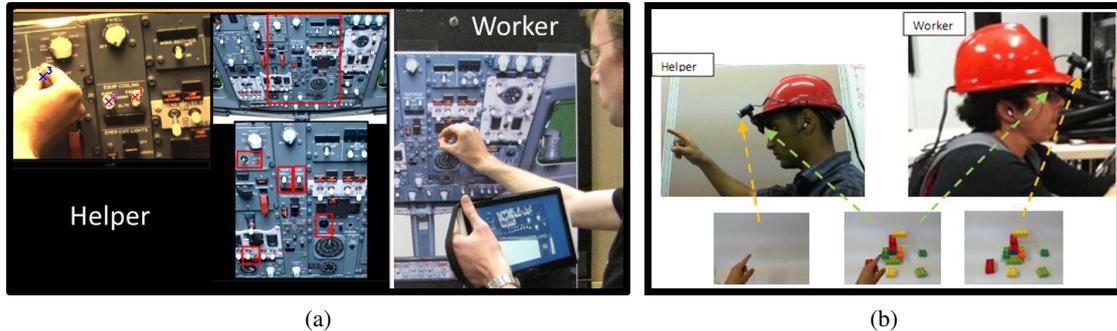


Figure 3.2: Manipulation guiding techniques. (a) An annotation-based approach presented in [Gauglitz et al., 2014b]. (b) A free mid-air gestures-based approach described in [Huang et al., 2011]

the proximity of visual cues according to the guided viewpoint.

3.2.2 Manipulation

The literature presents remote collaborative systems based on shared wall-sized screen allowing remote gestures guiding. The display creates a virtual window through the remote workspace, handling co-location with head-tracked stereoscopy [Towles et al., 2002], and even interaction with shared physical interface [Zillner et al., 2014]. Most applications targeting remote guiding for maintenance adds interactive features such as the ability to sketch guiding cue and to help navigation [Gauglitz et al., 2014a] with a correct 3D mapping into the real world while the agent is moving, as well as a viewpoint saving mechanism easing the remote expert's task [Gauglitz et al., 2014b] (cf. Fig. 3.2a).

Alem et al. developed many MR interfaces for remote collaborative maintenance based on shared point of view. HandsInAir [Huang et al., 2011] (cf. Fig. 3.2b) and MobileHelper [Robert et al., 2013] are examples of systems that provide gesture-based guiding for an agent helped by a remote expert. The agent wears a helmet with a camera that is streamed to the helper. Then, the helper acts with free mid-air gestures in front of a camera, and the helper's hands are merged to the output display of both users. Extensions using 3D capturing have been proposed with HandsIn3D [Huang et al., 2013c] and the 3D Helping Hands [Tecchia et al., 2012] systems. They add 3D handling of occlusions and virtual shadows and, thus, improve immersion and the sense of presence. Nevertheless, the displays are still not collocated with the real environment, and the interaction of the helper stays limited to moving the head around the position of the agent's head and to showing the gestures to perform without being able to interact with any object. In the following of this chapter, we aim to overcome these limitations by proposing techniques that improve available interactions for the expert and that decrease perception issues for both users, the expert and the operator.

3.3 Proposed approaches

This chapter focuses on providing efficient interfaces in order to enhance remote collaboration in MR. Particularly, we restrained our field of investigation on two remote collaborators and proposed new ways to represent distant workspaces and collaborator's activity.

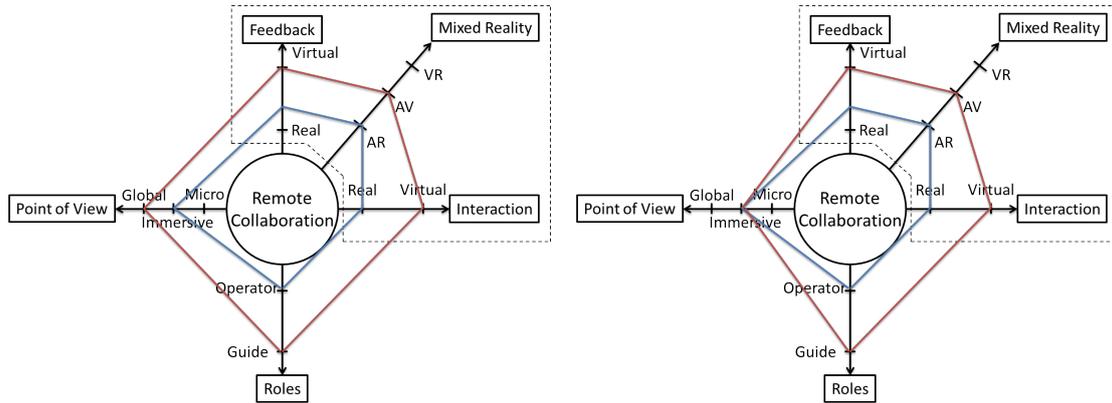


Figure 3.3: Description of our two approaches in terms of asymmetrical dimensions introduced in our systems. (Left) The HWD and WIM system (1) described in section 3.4. (Right) The fully collocated system (2) presented in section 3.5.

Here, we present our contributions in two distinct phases. First, the remote expert must be able to diagnose the distant workspace and furnitures, while guiding the agent to follow him/her in order to place the agent in the correct location. This navigation can take place in buildings containing corridors, complex room shapes and furnitures layout or complex system shapes in industry. Thus, we propose new guiding techniques for this navigation purpose and evaluate them. Synchronous **guiding techniques for remote navigation** can be divided into two families:

- **Static guiding:** the expert points a spot to reach; then the agent walks to this spot. It implies that the workspace perceived by the remote expert is currently containing the desired spot to reach (i.e. the expert can see the spot at the time he/she defines his/her guidance)
- **Dynamic guiding:** the expert continuously guides the agent to move in a certain direction. The expert can change the guiding direction at all time. This setting allows dynamic discovery of the workspace by the remote expert. It allows to guide the agent, until a spot initially not seen by the expert but dynamically discovered, thanks to the agent’s motion who can move the streamed camera.

Once the desired spot for performing the maintenance is reached, the **physical manipulation guiding** can begin.

Settings In our work, we explored the following settings in order to manage collaborative remote maintenance:

- **(1) Asymmetrical: HWD and WIM**
 - the agent is guided through a HWD;
 - the expert helps with a global view of the remote workspace.
- **(2) Symmetrical: Fully collocated**
 - the agent is guided through 3D collocated augmentations in his real field of view (using video see-through HMD);
 - the expert is immersed in 1:1 scale remote workspace (using HMD) and guides using real scale interactions.

Figure 3.3 illustrates the asymmetrical dimensions introduced in our both proposals. The first setting uses more simple interfaces and, then, turns to be more deployable, while the second one provides free-hand interactions for the agent and really immersed the remote expert in the maintenance workspace. However, both revealed advantages compared to the literature solutions.

First, concerning navigation guiding, we propose both guiding conditions (i.e static and dynamic) in setting (1), whereas in setting (2) the dynamic guiding condition is more relevant to investigate than the static one because of the 1:1 scale interactions of the expert that are not designed to point a spot out of reach.

Second, concerning manipulation guiding, most systems propose visual guiding cues such as sketching, notes or arrows. However, these interaction techniques are not natural to guide someone to perform a physical task. Indeed, compared with real life guiding in a local context, the expert usually shows, at least once, how to proceed to the operator; then the operator tries to reproduce what he/she just saw. Thus, to move toward this real life process, setting (1) makes it possible to display virtual clones of real objects, that the expert can manipulate, integrated into the operator's real world thanks to an AR display managing occlusions and shadows to reduce depth cues conflicts, which is much less ambiguous to understand. Moreover, setting (2) provides to the expert natural gestures at 1:1 scale as guiding interfaces, sharing the agent's viewpoint or in a face-to-face configuration, which can also be more efficient than guiding with verbal instructions. Our contributions using the setting (1) are presented in next section 3.4, and with the setting (2) in section 3.5.

3.4 Asymmetrical setting: HWD and WIM

The proposed system to perform remote guiding for collaborative maintenance is based on the streaming of the agent's camera to the expert. Then, the agent is helped through a Hand-Worn AR interface. Figure 3.4 describes our system instantiating our awareness model merging virtual and real spaces, then Figure 3.5 splits the previous figure in order to describe precisely what belongs to real spaces and virtual spaces.

In AR, from the operator side, some perception issues are generated because of the unhandled occlusions between real and virtual objects that create depth cues conflicts causing difficulties to perceive properly the relative depth. In our setting, we handle this feature with the use of a depth camera and 3D reconstruction of the real environment. Moreover, it enables the streaming of this 3D reconstruction to the expert in order to let him perceive the maintenance space in 3D. The expert can move his own PoV around the remote reconstructed workspace. Moreover, several interaction tools are provided to the expert who can:

- point items to the agent using a 3D ray;
- add/remove virtual arrows in the shared VE in order to add fix virtual pointers into the shared VE;
- segment parts of the reconstruction to create movable virtual objects;
- guide the agent using the navigation technique presented in section 3.4.2.

Figure 3.6 illustrates the system with screenshots of some proposed features.

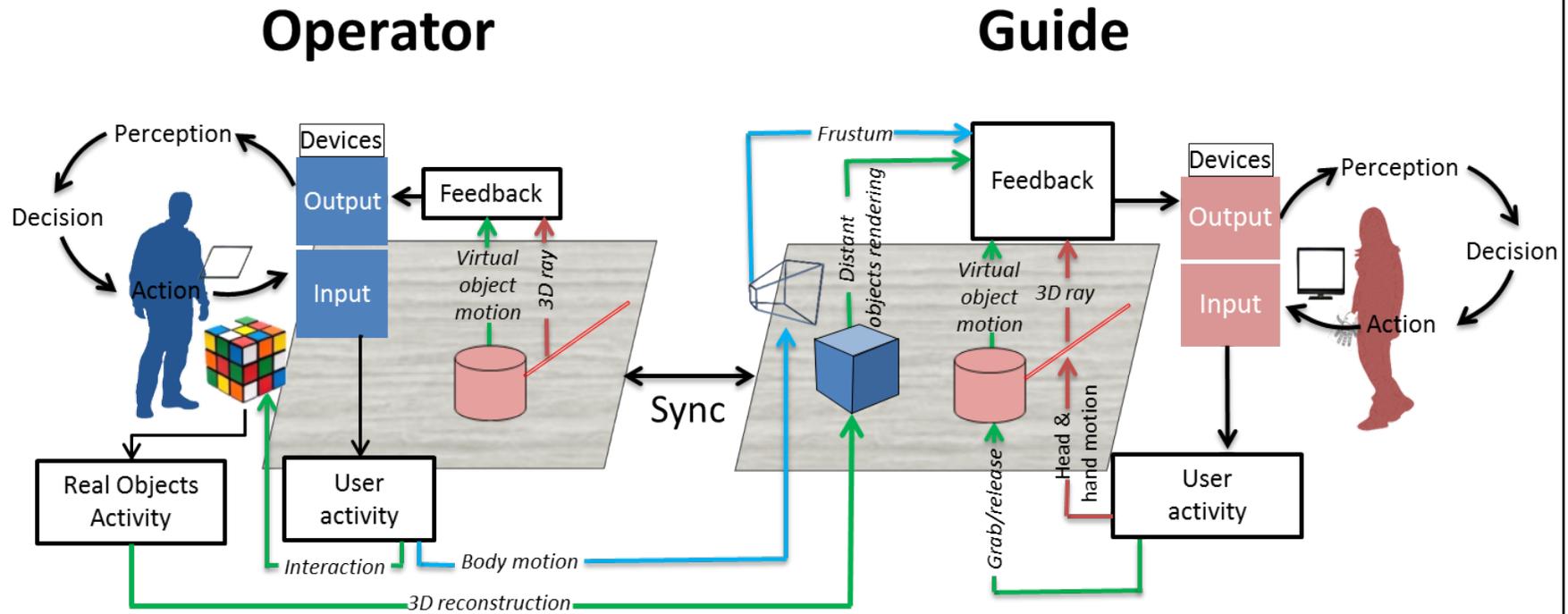


Figure 3.4: Description of our HWD and WIM system instantiating our awareness model and merging virtual and real spaces. A split version of this model is provided in Figure 3.5.

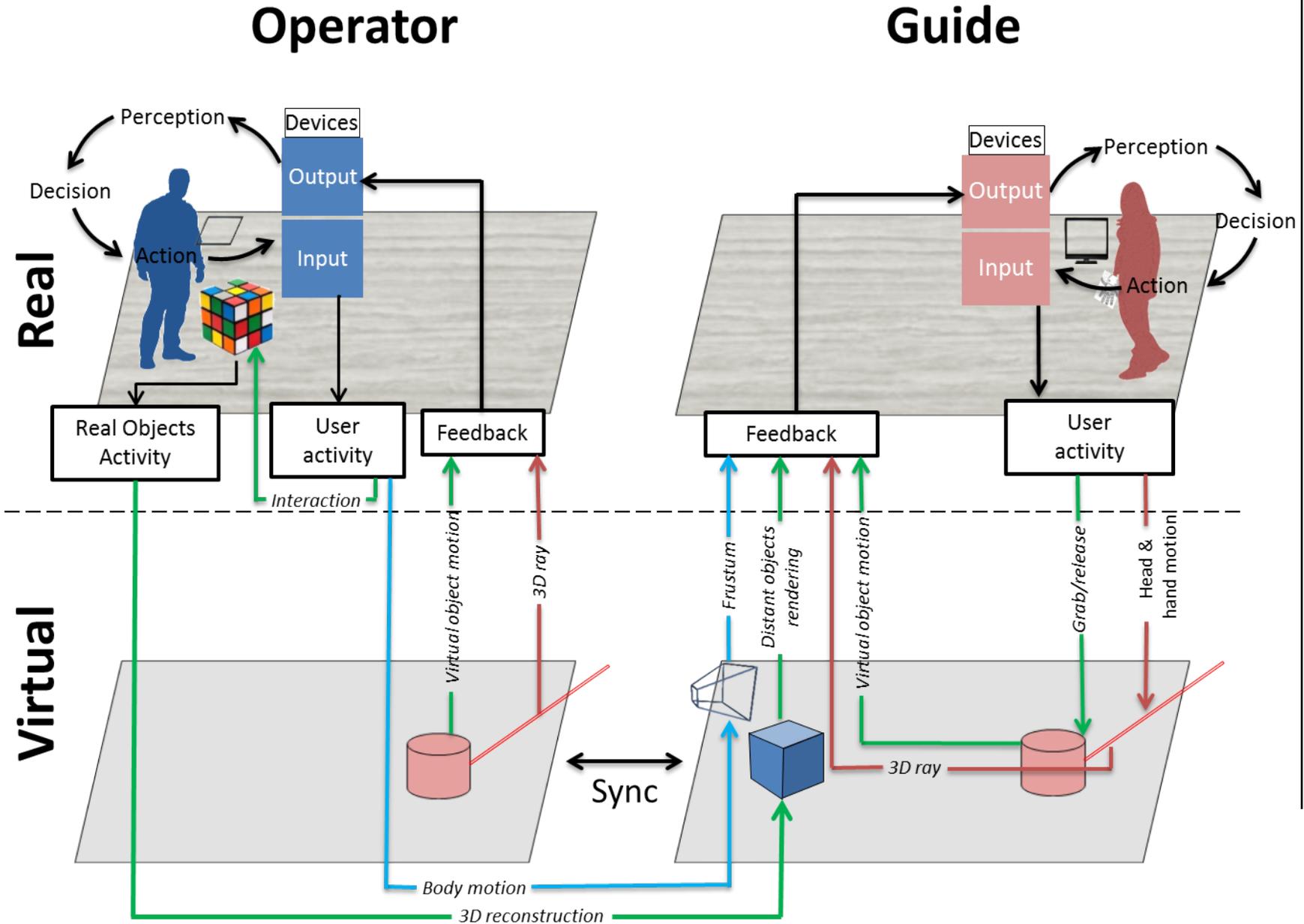


Figure 3.5: Description of our HWD and WIM system instantiating our awareness model.

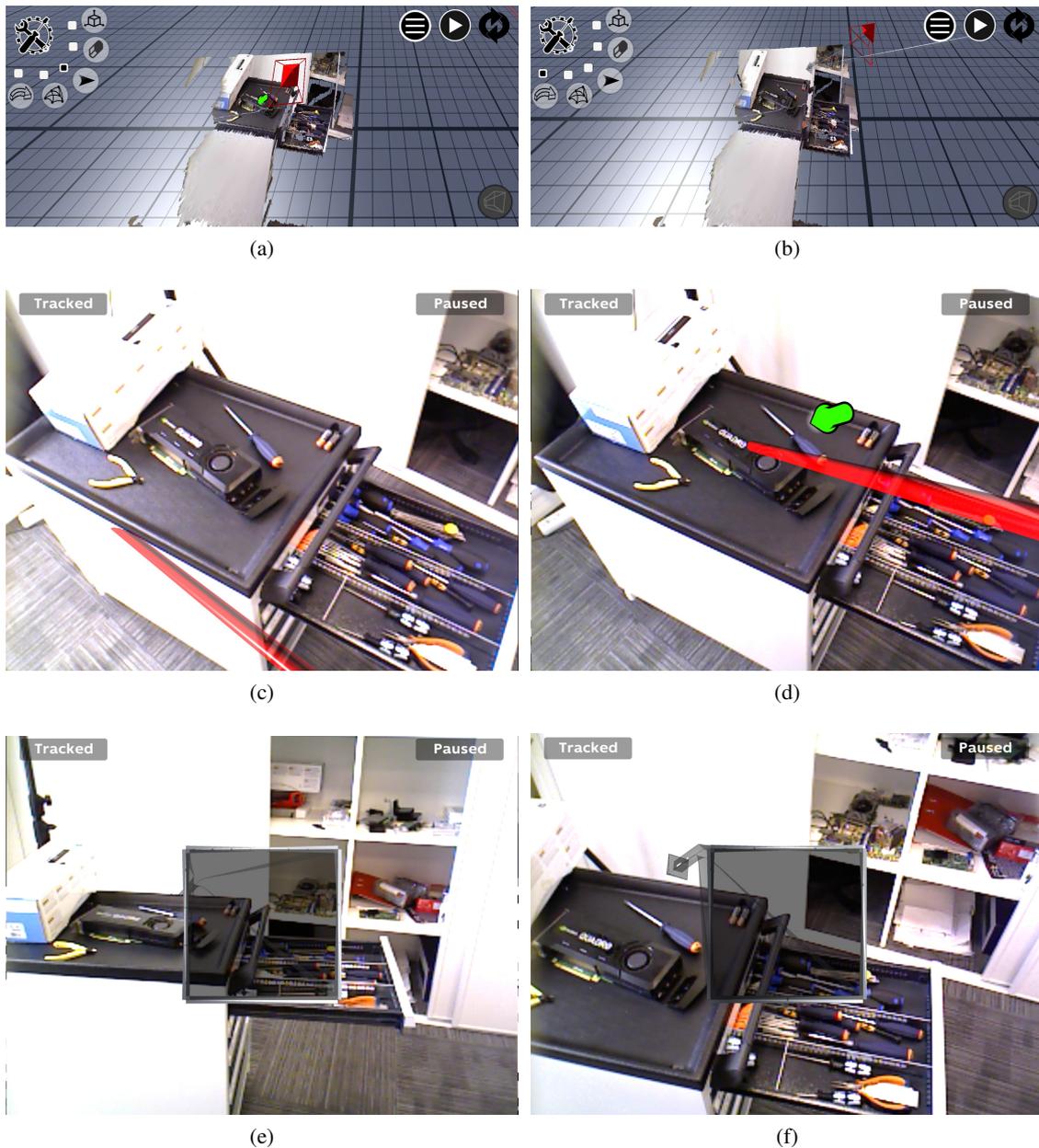


Figure 3.6: Screenshots of the HWD and WIM prototype. (a) and (b) Expert's view: the remote workspace is reconstructed in 3D and displayed on a stereoscopic screen with head-tracking. The red frustum represents the current operator's PoV. (a) The expert just pointed the graphic card using a static virtual arrow placed in the shared VE. (b) The expert is dynamically controlling the arrow used for navigation (cf. section 3.4.2). (c), (d), (e) and (f) Agent's view: the AR view mixes real and virtual content. (c) the expert's virtual 3D ray is occluded by the furniture thanks to the dynamic 3D reconstruction that provides a proper perception of the spatial layout of real and virtual objects. (d) The expert pointed the screwdriver with the static green arrow and indicates where to use it on the graphic card with the red 3D ray. (e) and (f) show how occlusion between virtual and real objects can be useful in some situations, for instance when pointing hidden spots. This navigation guiding technique is explained in section 3.4.2.



Figure 3.7: Illustration of the asymmetrical (HWD and WIM) prototype. The agent on the left is guided by the remote expert on the right.

3.4.1 Implementation

Figure 3.7 illustrates our prototype in practice. The developed AR device prototype is a wireless tablet PC (Microsoft Surface Pro 3) coupled with a Google Tango device used to compute 6 DoFs self-positioning in real space (using inside-out tracking based on SLAM with wide field of view fused with IMU data) and equipped with a high resolution USB-powered depth camera (Asus Xtion Pro). This setup handles real-time 3D reconstruction from a 640x480 depth map provided by the depth camera as well as marker-free tracking. Figure 3.8 details this setup. The whole setup needs a registration phase to define the rigid transformation between the fish-eye tracking camera and the RGB camera used for AR overlay. It is achieved using OpenCV camera calibration method based on a chessboard image captured by both cameras at the same time. Then, the rigid transformation is the difference between the two computed poses according to the chessboard. Notice that the Tango tracking tablet is also equipped with a depth camera, but this tablet does not have enough processing power to perform real-time 3D reconstruction at high resolution (its depth camera produced a sparse depth map of ~ 10000 points, i.e. resolution $\sim 100 \times 100$). The current expert's setup is a zSpace that provides head-tracked stereoscopic display and a 6 DoFs stylus for interaction.

3.4.2 Navigation guiding

The main advantage for the expert with this setting is the ability to dynamically discover the remote workspace based on his/her own guidance. Thus, the expert can build his/her own awareness of the remote workspace and operator's activity using this *dynamic 3D reconstruction for discovery* feature.

In our prototype, the occlusion handling for the operator's AR display is useful to provide a proper perception of depth for virtual objects integrated into the real world, but it could also hide some of the visual guiding cues controlled by the remote expert. Furthermore, if the expert wants the agent goes in a spot situated on sides or even behind the agent's field of view, it can be tricky to efficiently perform the navigation guiding. Thus, we propose a navigation technique that handles a visual continuity between the agent's HWD and the spot to go. It is composed of a 3D arrow made with spline controlled by three control points with 6 DoFs (cf. Fig. 3.9).

On the agent side, the base of the squared arrow (i.e. the base control point) is fixed relatively to the HWD AR display. Thus, the agent must move it to align this square with the square formed by the end of the arrow. Once done, a proximity sensor is displayed and flicks according to the distance to the final location (using position and orientation differences). The final location is

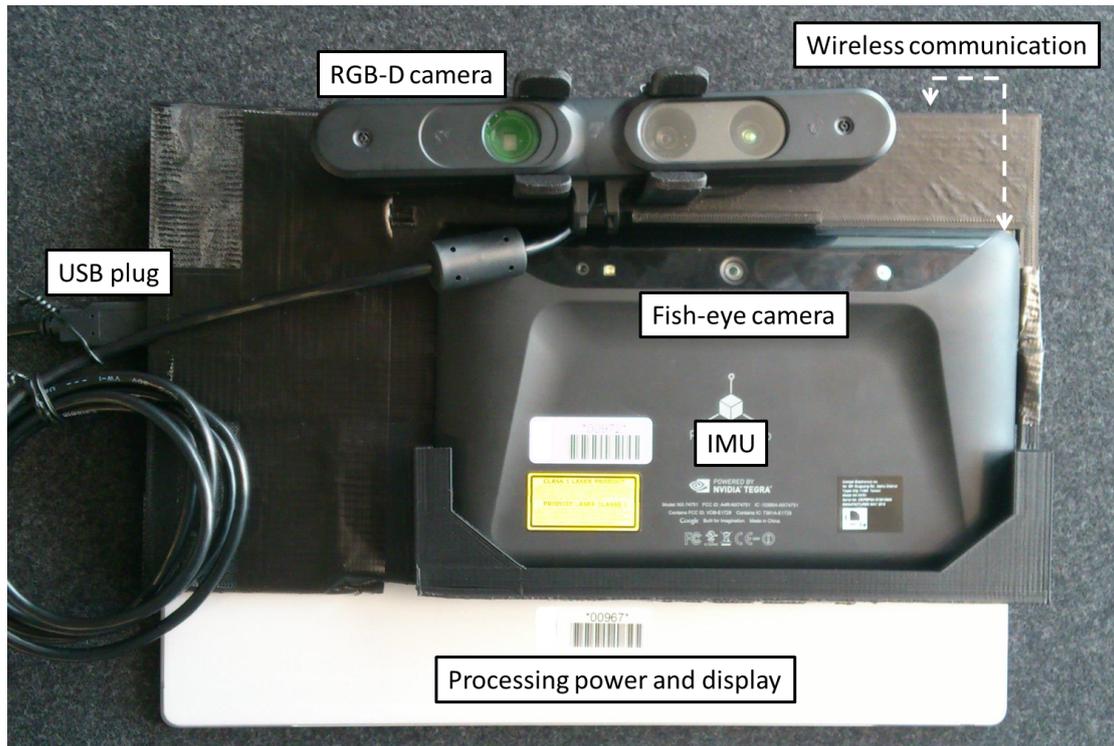


Figure 3.8: The back of our AR tablet prototype. The Google Tango tablet uses sensor fusion from SLAM with fish eye camera and IMU to provide a 6 DoFs tracking. The RGB-D camera is a Asus Xtion powered with USB. These two devices are mounted thanks to a 3D printed support to a Microsoft Surface Pro 3 (Intel CoreI7 with HD Graphics 5000 chipset) providing processing resources as well as a larger display.

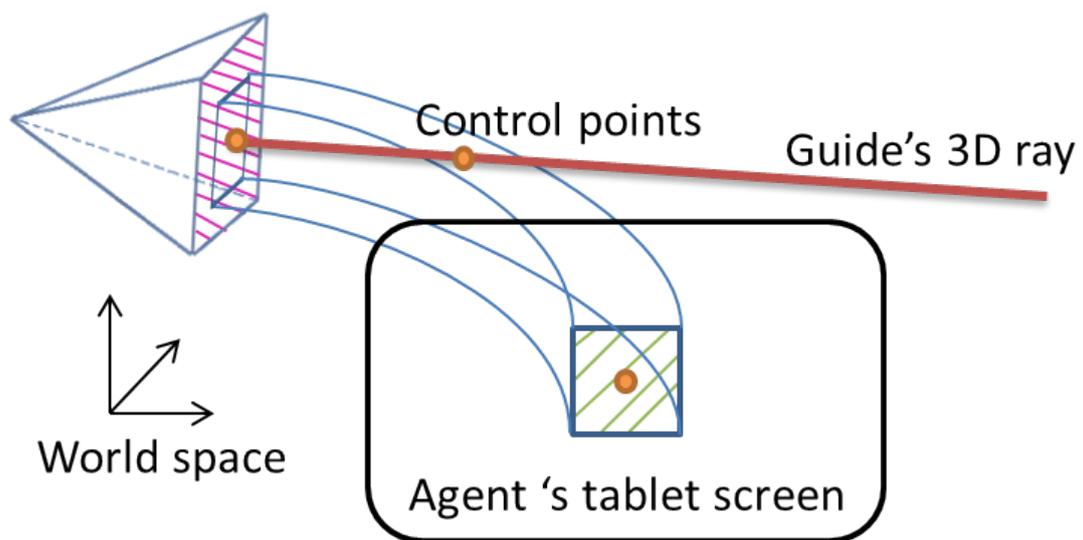
correct as soon as both squares (i.e the base and the final squares) are superposed.

On the expert side, he can guide the agent using two modes. The static mode allows to define a spherical volume (using the 3D ray) in the workspace, then the system computes locations of two final control points, using the 3D ray to define orientation, to place the guiding arrow in a proper way. When the agent will reach the pointed spot, his camera will contain the defined volume in his viewing frustum. Figure 3.10 illustrates this process. The dynamic mode fix the two final control points to the 3D ray used as a virtual interaction tool by the expert to perform the remote guiding. The first control point is at the ray extremity, and the other one is at a variable distance from the ray extremity depending on the desired spline curvature. In dynamic mode, the expert continuously controls the end of the squared guiding arrow and can refine his guidance while discovering the remote workspace.

3.4.2.1 Experiment

We conducted an experiment on the agent side in order to evaluate the effect of the visual continuity aspect introduced with our navigation technique.

Protocol & Hypotheses We compared reaching performances to a statically defined target using our technique (mode 1) versus a simple guiding technique based on a compass and a 3D



(a)



(b)

(c)

Figure 3.9: Squared guiding arrow principle controlled by the expert's 3D ray in dynamic mode. (a) Scheme of the control principle constrained by three control points. (b) The agent's display showing a PoV to reach dynamically controlled by the expert's 3D ray (seen on the top right of the screenshot). (c) The agent reached the pointed spot; the base and the end squares of the arrow are aligned and almost superposed; the proximity sensor (displayed as a reticule) is flickering.

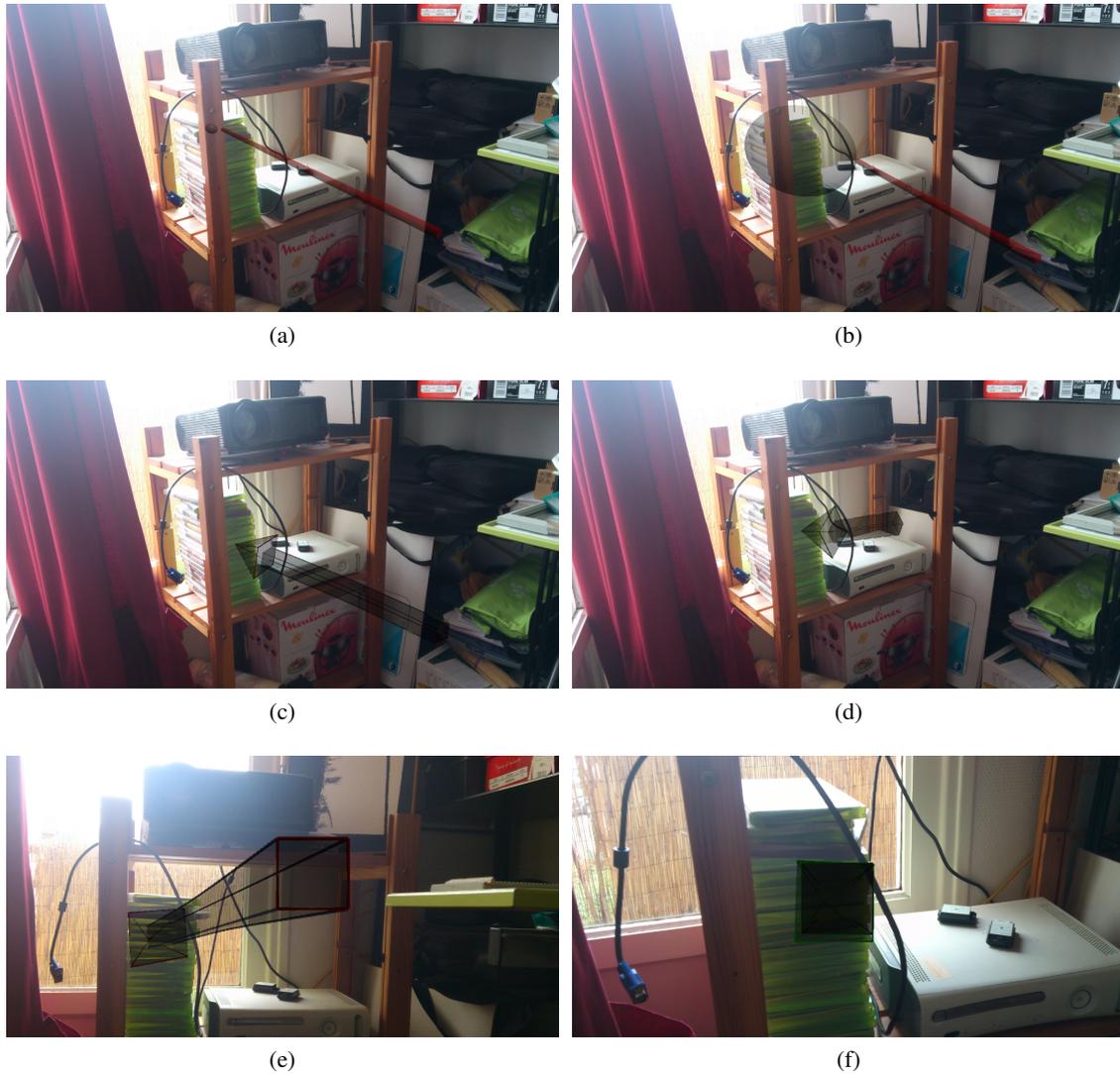


Figure 3.10: Illustration of the static mode for navigation guiding. (a), (b), (c) and (d) are exocentric PoVs from the agent's workspace used to explain the expert interactions, while (e) and (f) are actual screenshots of the agent's display. First, the expert defines a virtual sphere using the 3D ray in (a) and (b). Then, the system computes the location of the two final control points using the sphere as bounding volume and the ray for their respective orientation in (c) and (e). Last, on (d) and (f), the agent reaches the correct spot pointed by the navigation technique; it has been computed in a way that the camera FoV, worn by the agent, contains the initially defined spherical bounding volume.

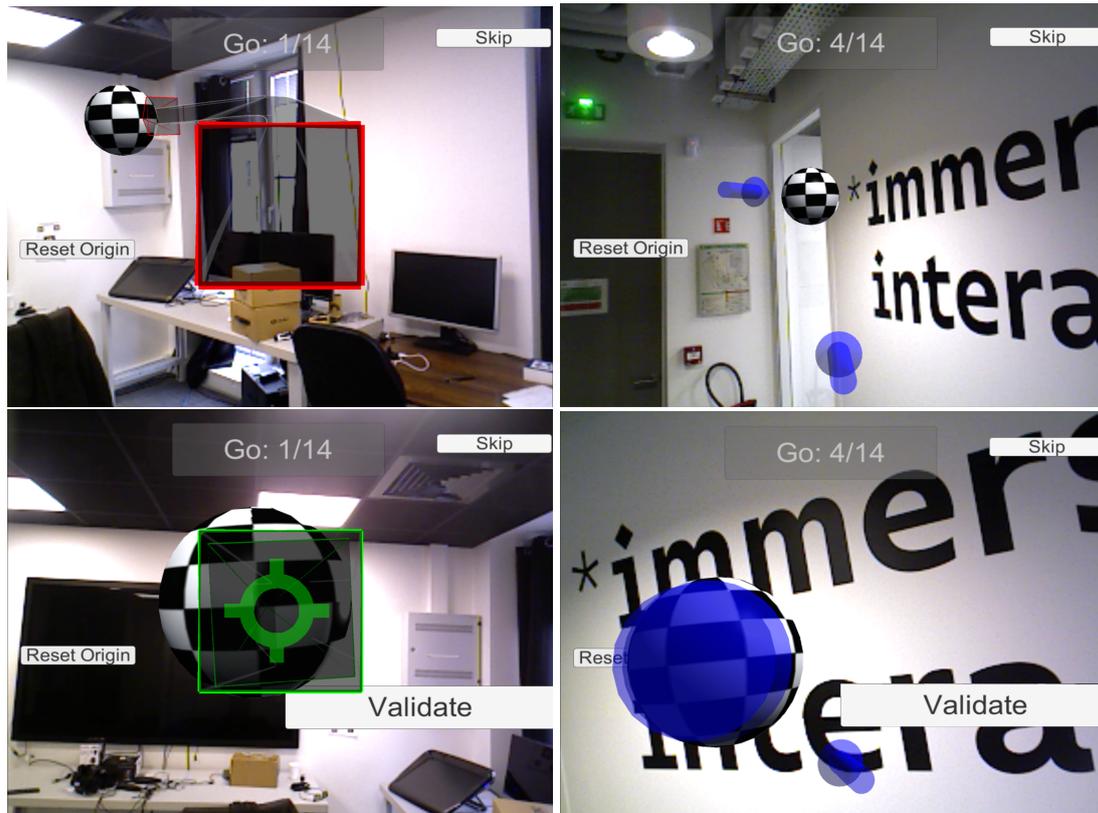


Figure 3.11: The two guiding techniques compared in the study. The squared guiding arrow on the left, and the pointer with compass on the right (the compass is the little cylindrical arrow on the bottom of the viewport). The checkered sphere is the target on which subjects must reach a specific PoV. On the bottom line, the user reaches the pointed target and the PoV is aligned enough with the desired PoV to validate the final precision.

cylindric arrow¹ (mode 2). In mode 2, the compass is always pointing into the direction of the target. It is useful to find its location when it is hidden by real furnitures or walls for instance. Once the target is visible in the field of view of the agent camera, the cylindric arrow is used to indicate the exact pointed PoV to reach in order to have the desired perception on the target.

We chose a static guiding condition because it allows us to remove the potential bias that could have been generated by an expert's interactions dynamically defining a target to reach. In order to follow the superposition of the squares principle used in mode 1, the correct location in mode 2 is also reached when the bottom circle is superposed with the big ring of the cylindric arrow. Figure 3.11 illustrates the two compared techniques and the alignment condition.

To begin, the subjects tested both guiding modes twice, then they performed 5 target reaching iterations per mode (cf. Fig.3.12). Between each iteration, subjects should come back to the original position of the floor in order to launch the next one. The mode order has been counter-balanced between all the participants.

We collected completion time until the subjects reach a threshold close to the target location (in terms of position and orientation), then they could take time to validate the final precision as

1. The compass is inspired from [Nguyen et al., 2013] and the cylindric arrow from [Tonnis et al., 2005]

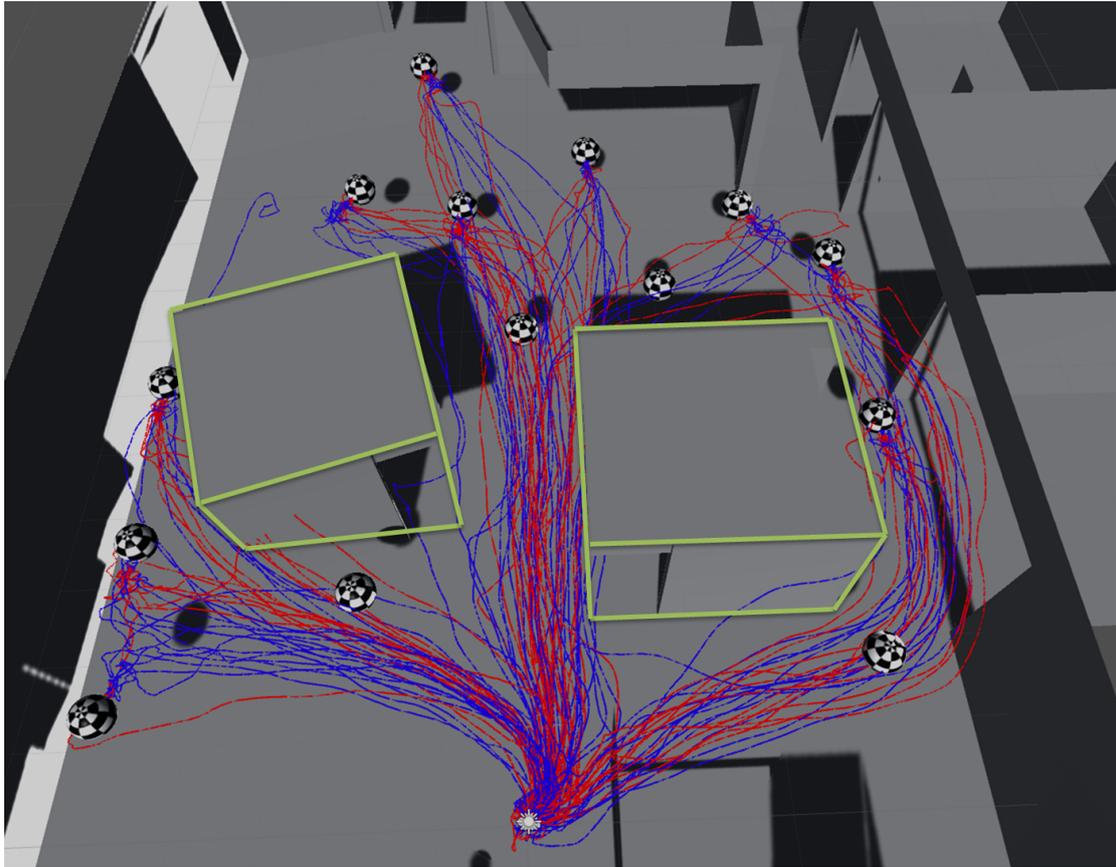


Figure 3.12: The targets location into the floor, and the participant's paths recorded during the experiment (blue: mode 1; red: mode 2. The green outlines represent the walls of two rooms present in the floor.

close as possible as the perfect location indicated by the current guiding metaphor (i.e perfect superposition of squares in mode 1 or circles in mode 2).

Our hypotheses are:

- H1: Completion time is smaller using mode 1 than mode 2 because the continuity aspect decreases the ambiguity of the guiding metaphor compared to a compass that points into the void.
- H2: The precision, especially in terms of orientation, is better using mode 1 than mode 2 because squares superposition allows a better 3D orientation perception than circles superposition that does not contain helpful angular visual cues.

Results We collected data from 14 subjects aged from 22 to 54 ($mean = 27.9, sd = 7.9$). Illumination conditions sometimes disturb the vision-based tracking performed by the Tango device, and thus we discard these iterations ($\sim 15\%$) from the analyzed results. Figure 3.13 illustrates the results. Completion time is significantly smaller in mode 1 ($M = 21.85, sd = 140.62$) than in mode 2 ($M = 27.92, sd = 344.11$): $F(1, 117) = 4.69, p < .05$. Concerning positional precision, we found no significant difference between the two modes. However, orientation precision is better using mode 1 than mode 2. In the approach phase (i.e at the time the system validates the target reaching), subjects reached targets with a significant better angular pre-

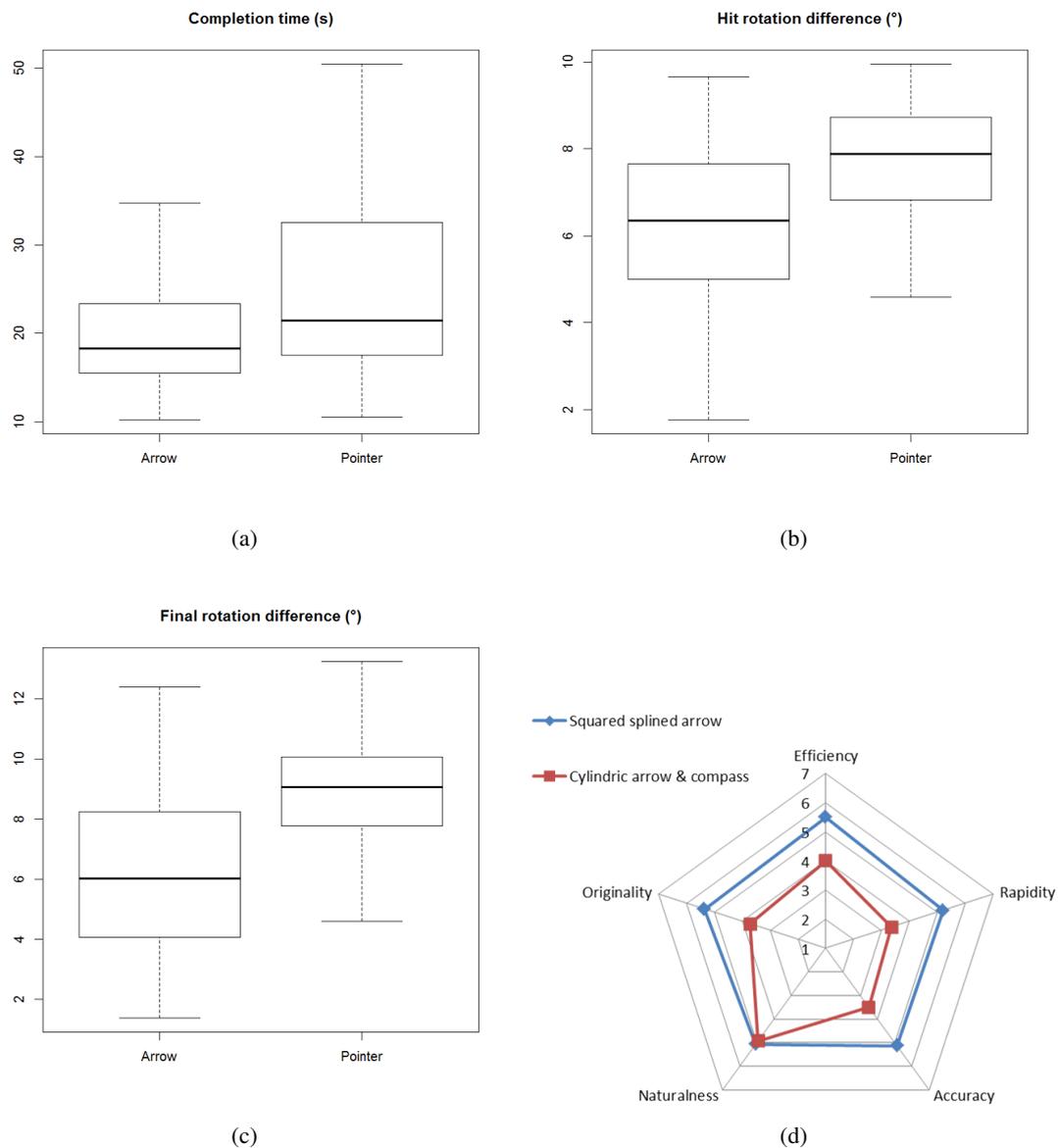


Figure 3.13: (a,b,c) Boxplots of collected quantitative results. (d) Summary of qualitative appreciations.

cision in mode 1 ($M = 6.02, sd = 4.66$) than in mode 2 ($M = 7.63, sd = 3.30$): $F(1, 117) = 18.77, p < 10^{-4}$. The same way, concerning the final angular precision (without taking into account the roll angle), mode 1 ($M = 6.19, sd = 7.38$) is significantly more accurate than mode 2 ($M = 8.92, sd = 8.18$): $F(1, 117) = 28.28, p < 10^{-6}$.

Last, concerning qualitative appreciations, subjects prefer mode 1 than mode 2 for every criterion (cf. Fig.3.13d).

Discussion We validated both our hypotheses. First, H1 proves that our technique, based on a continued squared arrow driven by a spline, is easier to follow compared to a simple direction

shown by a compass. This generates a faster reaching to the remotely pointed target, and thus improves the whole process of the distant guiding. Second, H2 demonstrates that the use of squared shapes is much more efficient in a 3D space than circled shapes. Indeed, squares convey angular informations contrary to circles. These angular visual cues are very important to provide an efficient way to reach specific orientations in a 3D space. Obviously, the roll angle can not be defined using the circles superposition principle. With our technique, this roll angle is handled thanks to the third control point attached to the agent's display and the continued spline that defines the whole guiding arrow. Moreover, pitch and yaw are also easier to perceive because the deformations of squares seems to carry more 3D informations, and thus is easier to perceive than circles deformations that create slightly ellipsoid shapes.

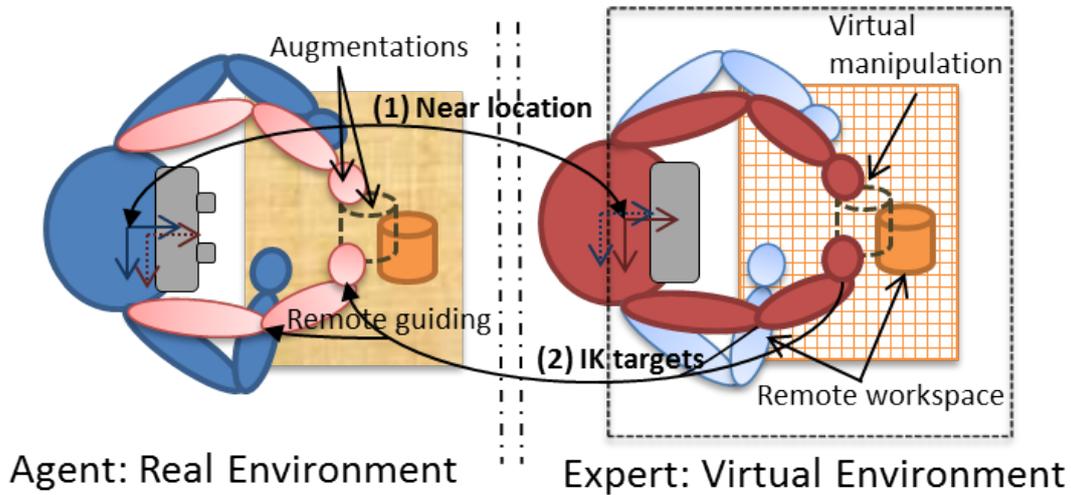


Figure 3.14: A conceptual top view of the *Vishnu* paradigm.

3.5 Symmetrical setting: Fully collocated

Our fully collocated system aims to overcome limitations encountered with remote maintenance application similar to our asymmetrical (HWD and WIM) prototype. Thus, it must satisfy some new constraints:

- the agent must be able to perform free hand-gestures, and to perceive augmentations integrated to the real space controlled by the expert, using a head-worn display;
- the remote guiding gestures must be easy to map in the agent’s real 3D world;
- the expert must be collocated in the distant workspace in order to have a proper viewpoint on it regarding the scale of the scene and to ease his interactions;
- the expert must be able to interact with virtual objects that must be seen by the remote agent.

We introduce *Vishnu*, a novel interaction paradigm that proposes to add two virtual arms to a local agent controlled by a remote expert. With these two additional arms coming out of the agent’s own shoulders, the remote expert can use them as interaction tools and show the exact gestures and actions to perform, as illustrated in Figure 3.14. Therefore, interaction is more natural for the expert, and it is easier for the agent to understand the remote guiding instructions. Moreover, it provides the guide with the ability to share the exact operator’s PoV which can improve his/her awareness about the operator’s activity compared to an asymmetrical setting in terms of PoV, such as the HWD and WIM prototype presented in previous section 3.5. Figure 3.15 instantiates our awareness model, merging virtual and real spaces, for this proposal in a face-to-face situation (thus, without the expert sharing the agent’s PoV); and figure 3.16 provides a more precise description of virtual and real contents for both users.

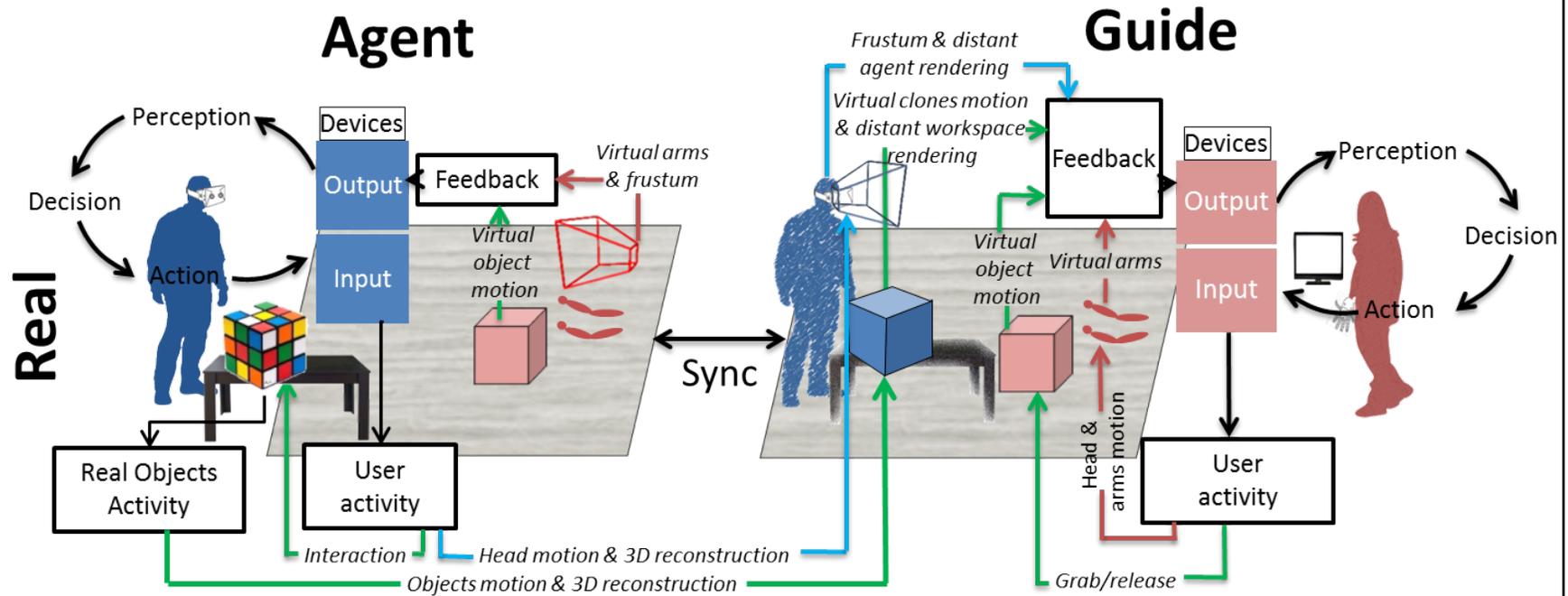


Figure 3.15: Description of our fully collocated system instantiating our awareness model and merging virtual and real spaces. More explanations and a more precise description that splits virtual and real spaces contents is provided in figure 3.16.

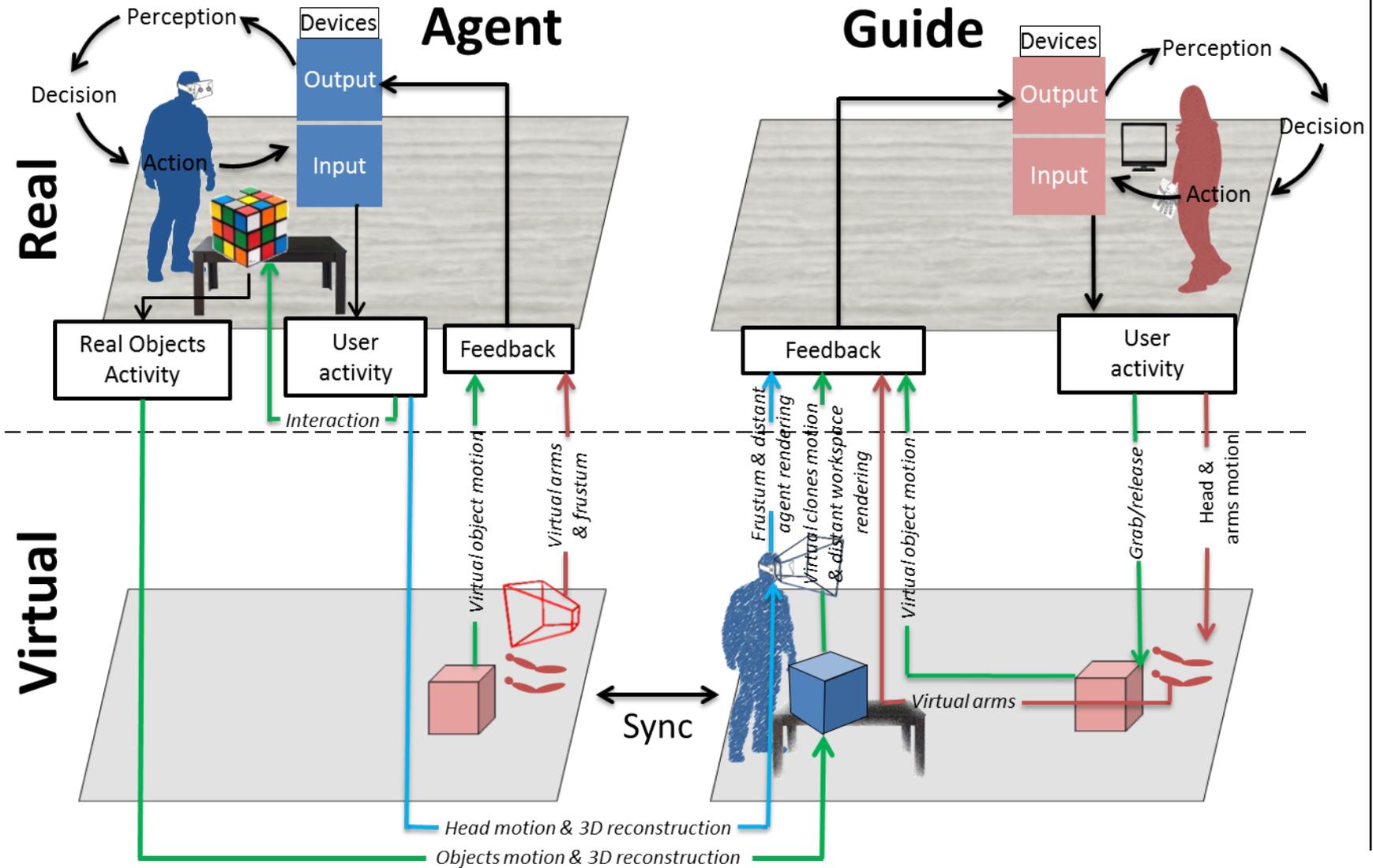


Figure 3.16: Description of our fully collocated system instantiating our awareness model. We did not enable the *Vishnu* paradigm in this description (i.e the expert does not share the agent's PoV) for clarity purpose. In the actual implementation, *Vishnu*'s arms would be transparent in this situation because the expert is too far from the agent's location, as explained in section 3.5.1.1 (except in a special mode that allows to disengage from the *Vishnu* paradigm).

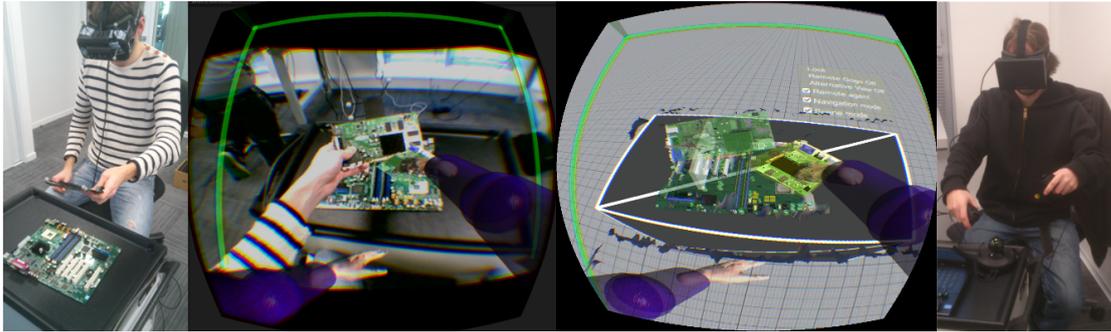


Figure 3.17: Illustration of the system and the viewpoints of the agent (left) and the expert (right) in a motherboard assembly scenario.

3.5.1 System design

The main idea of *Vishnu* is to add two virtual arms to an agent, coming out of the agent's own shoulders. Thus, the agent can see his/her two own real arms in addition to the two additional virtual arms that are controlled by a remote expert, who can use them as interactive guidance tools. Figure 3.17 illustrates this paradigm that could be used in many applications: industrial maintenance, procedure learning and sports training. It is based on a bi-directional communication (cf Fig. 3.14). The expert's virtual location must be close to the agent's real position and orientation (1). The way the expert navigates in the shared VE is described in subsection 3.5.1.2. Then, the expert's virtual arm gestures define Inverse Kinematics (IK) targets used to control the virtual arms of *Vishnu* for the agent (2). With *Vishnu*, we overcome the limitations highlighted in the related work for both the expert and the agent in order to improve their sense of co-presence and to decrease their cognitive load due to a de-localized display and interactions.

In the following, we present the features of our system, including the *Vishnu* paradigm in addition to others that provide a highly usable system.

3.5.1.1 Agent Features

First, a main goal of our system is to provide the agent with a true AR display enabling a proper perception of the augmented environment. To achieve this, we need to dynamically reconstruct the real environment in 3D in order to handle occlusions of real objects by virtual ones (cf. section 3.5.1.3 for implementation details). Moreover, this allows the handling of virtual shadows (i.e virtual objects are able to cast shadows on real objects) that increase the sense of presence and ease the perception of depth. Furthermore, the agent's head is tracked to provide a collocated stereoscopic display.

The *Vishnu* paradigm is based on an IK algorithm [Tolani et al., 2000] that controls the virtual arms with virtual targets for the elbows and hands while the shoulders are fixed regarding the agent's ones. This makes sense only if the agent is close to the expert's virtual location. Otherwise, the virtual targets are too far from the hands and elbows controlled by the expert, and the remote gestures are not relevant anymore. Thus, the expert is represented with a viewing frustum that changes its color from red to green according to the distance between the two users. This provides a basic feedback to the agent that enables him/her to place him/herself correctly when necessary. In the same way, the virtual arms smoothly disappear when the agent moves

away from the expert's location.

3.5.1.2 Expert Features

The strongest limitation found in previous work is the limited interactions of the expert. Our system provides a VR setup for the expert that enables him/her to interact in the shared VE in a richer way. In addition to the control of his/her viewpoint, the expert controls two virtual arms that can interact with virtual objects. Thus, the expert can grasp and manipulate virtual objects that are displayed as augmentations on the agent side (cf. Fig. 3.17).

The expert's arm parts (i.e. shoulders, elbows and hands) are tracked and used to control their virtual representations for proper kinematics between the expert's gestures and his interactive tools. The manipulation of a virtual object is achieved using an interaction technique based on a virtual hand metaphor [Achibet et al., 2014]. The *Vishnu*'s hands close around the virtual objects in a natural way using a simple grasping algorithm. Moreover, another interaction allows to use the virtual hands to point something. Instead to grasp an object, the hand closes without the index finger as illustrated in Figure 3.18.

Concerning navigation, the expert can freely move in the VE. But, to guide the remote agent with his/her gestures, the expert must be close to the agent's location. Thus, two interactions are possible:

- smoothly bringing back his/her location and orientation to the remote agent's current one.
- locking/unlocking the automatic following of the remote agent's head by the expert's virtual PoV.

When the expert shares the viewpoint of the agent, a navigation technique can be enabled. This technique allows the expert to guide the agent, using the *Vishnu*'s arms, in order to place the agent correctly to begin the manipulation task (cf. section 3.5.3).

Last, the 3D reconstruction of the remote agent and workspace can be enabled depending on current awareness needs. Especially in the diagnostic phase, the expert can stand face to the agent and freely explore the remote workspace in order to find the correct place to perform the maintenance task (cf Fig. 3.18).

3.5.1.3 Implementation

Figure 3.19 illustrates our implemented setups. For the agent, the 6 Dof tracking of his head and of real objects (initially overlaid with virtual clones) is performed with an OptiTrack V120:Trio™ IR tracking system. The 3D reconstruction is based on Kinect™. The AR display is a video see-through HMD made with an Oculus Rift™ equipped with two front stereo cameras in order to provide AR with wide field of view.

For the expert, the current system uses a Razer™ Hydra. Using one controller for each hand allows a 6 Dof tracking and provides buttons to interact and joysticks to navigate. The tracking of the expert's shoulders and elbows is based on a Kinect™. Last, the display can be done with a desktop screen or with an HMD such as an Oculus Rift™ to handle head-tracked 3D stereoscopy.

Applications are based on Unity3D™ C# scripting, and CG shaders handling Kinect-based real-time 3D reconstruction (also used for AR virtual occlusions and shadows). In future versions of our system, this 3D reconstruction of the agent's workspace should be done based on the stereo cameras, and the optical tracking system replaced by image-based tracking (i.e SLAM and/or 3D models-based). Moreover, the expert's VE should not need a pre-modeling phase, and

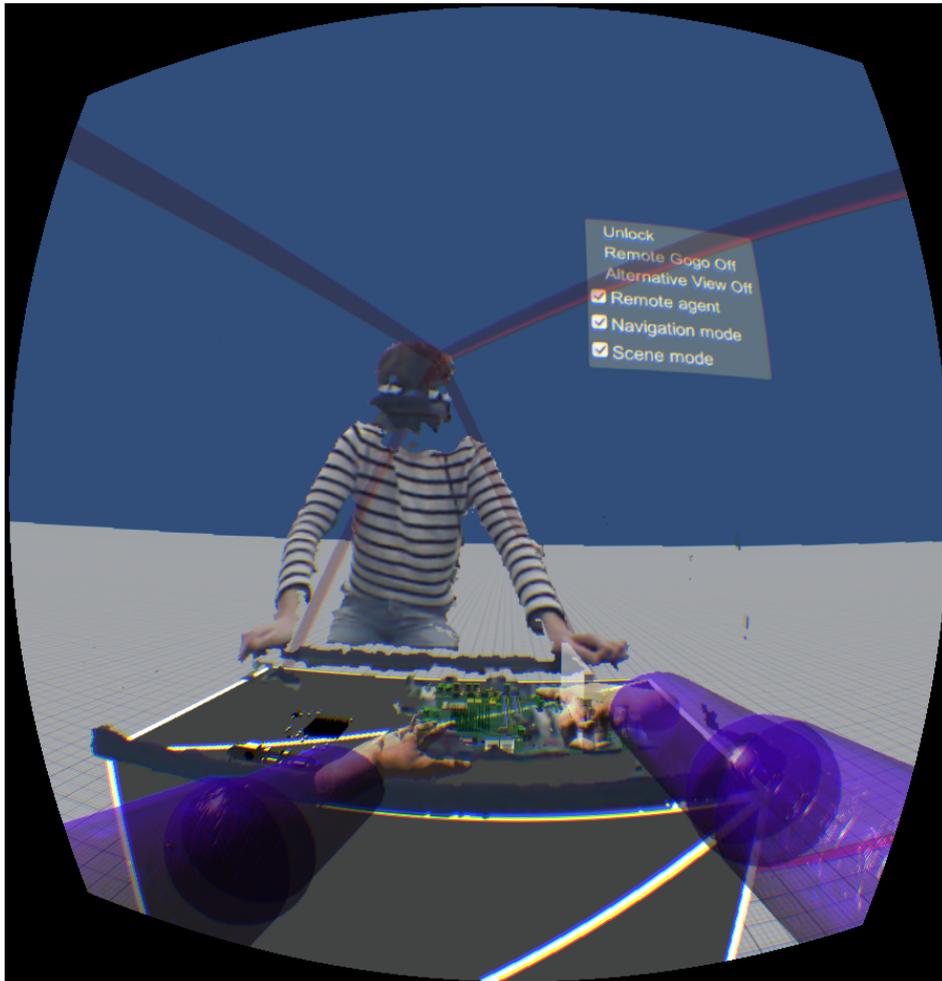


Figure 3.18: An illustration of the remote expert’s viewpoint standing in front of the distant agent 3D reconstruction.

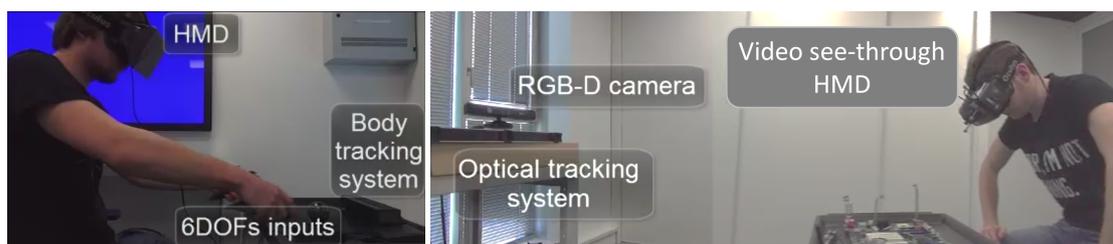


Figure 3.19: Implemented setups: the expert on the left and the agent on the right.

could be dynamically generated based on the 3D reconstruction coupled with interactive meshes segmentation and fusion [Dou and Fuchs, 2014].

3.5.2 Pilot user study

We ran a pilot user study with few subjects that aims to evaluate the effectiveness and usability of our system in a simplified scenario: the subjects are seated and only the selection

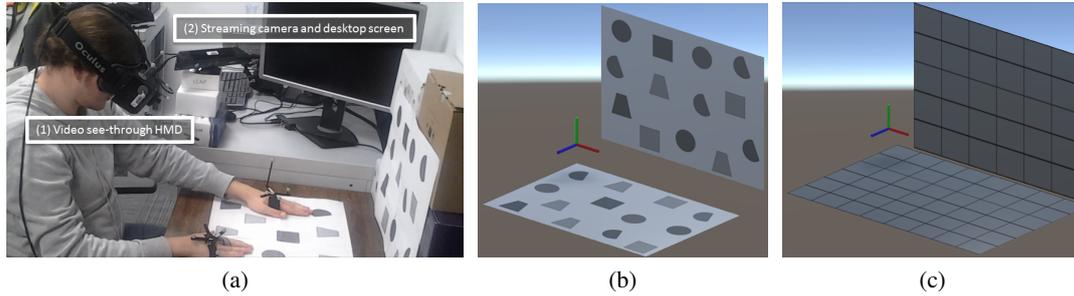


Figure 3.20: (a) Illustration of a subject performing the pilot user study in mode 1 with the first targets' layout L1. The virtual targets' spaces ((b): first layout L1; (c): second layout L2), reproduced from the real agent's workspace, used in the expert's VE.

task is evaluated, not the manipulation one. Subjects were asked to compare our system (mode 1) versus a basic one (mode 2) based on a desktop screen on the agent side and a fix camera streamed to the remote expert. From this point of view, the expert could sketch in 2D on the camera stream in order to guide the agent in his task. Subjects were the agent, while the expert was the same trained experimenter. Indeed, the expert's role needs a learning process, while the agent only has to follow remote instructions. The task simulated a physical bimanual selection by touching simultaneously two different targets (cf Fig. 3.20a), one with each hand. Since in mode 2 simple sketching provides no way to distinguish which hand the agent should use to select a given target, the expert used a color code to indicate which hand to use. No verbal communication in both modes was allowed. Each subject had to accomplish this task 20 times per mode: 10 times replayed from a pre-recording guiding phase performed by the trained experimenter, and 10 times with live manual guiding performed by the same trained experimenter. We stated hypotheses:

- H1: The subject's task completion from the completion of the guidance is faster in mode 1;
- H2: Mode 1 eases the mapping process between the guiding instructions and the physical task interaction space.

We ran this experiment twice with the same subjects' panel and conditions. First, L1 with a simple random targets' layout (cf. Fig. 3.20b), then L2 with a more regular and dense targets' layout (cf. Fig. 3.20c) that seems more complex because of less visual and geometric cues. For each iteration, the expert was pointed two targets to be selected (one for each hand) amongst all the available targets; and the subject should follow these instructions to select the designated targets with the correct hands. Figure 3.21 and 3.22 illustrate respectively the subject's PoV in mode 1 and in mode 2. To control the experimental variables, the targets designated by the expert were ordered in a counter-balanced way.

3.5.2.1 Results

Results have been collected from 11 subjects aged from 23 to 40 ($mean = 30, sd = 5.7$). Participants have been selected according to their acceptability of the video see-through HMD (mostly induced by their previous uses of VR HMD) in order to minimize its effect in mode 1 with respect to mode 2.

For L1, boxplot of figure 3.23a shows a significant difference between both modes con-



Figure 3.21: Illustration of a subject's PoV in mode 1. The Vishnu's arms are designating targets. The red and blue cube represent the tracked hands of the subject.

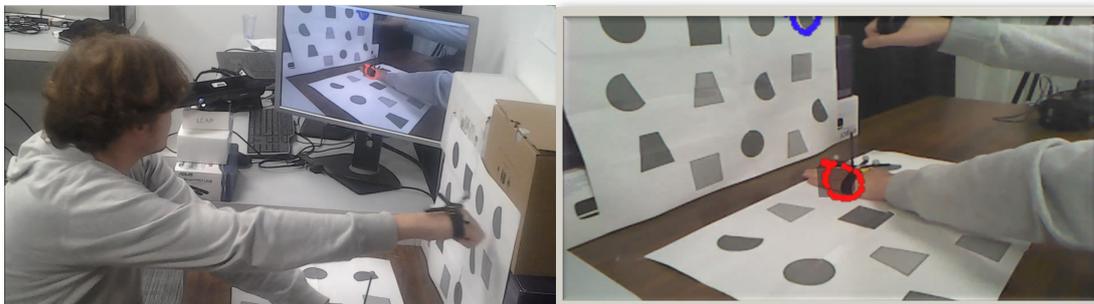


Figure 3.22: Illustration of a subject's PoV in mode 2. (Left) A global view of the subject and the workspace. (Right) A screenshot of the screen the subject is looking at in order to follow the instructions for the targets selection.

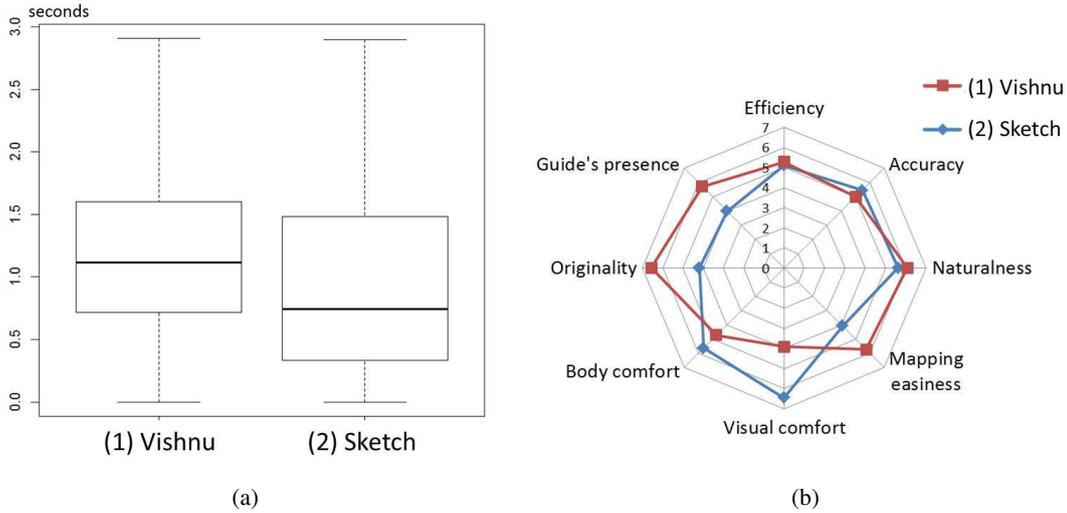


Figure 3.23: Results for the first layout L1. (a) Boxplot of the difference between the guidance completion and the subject's task. (b) Synthesis of qualitative results.

cerning the time difference between the guidance completion and the subject's task completion ($F(1,432) = 3.93, p = 0.047$). We can also notice that we found no significant difference between the replayed iterations and the manual ones.

Concerning the qualitative results (cf Fig. 3.23b), mode 1 is preferred in terms of mapping easiness, originality as well as guide's presence. Mode 2 offers a better comfort even if the desktop screen generates much more head movement than mode 1 (respectively a yaw angle range of 195° versus 103°), as illustrated on figure 3.24.

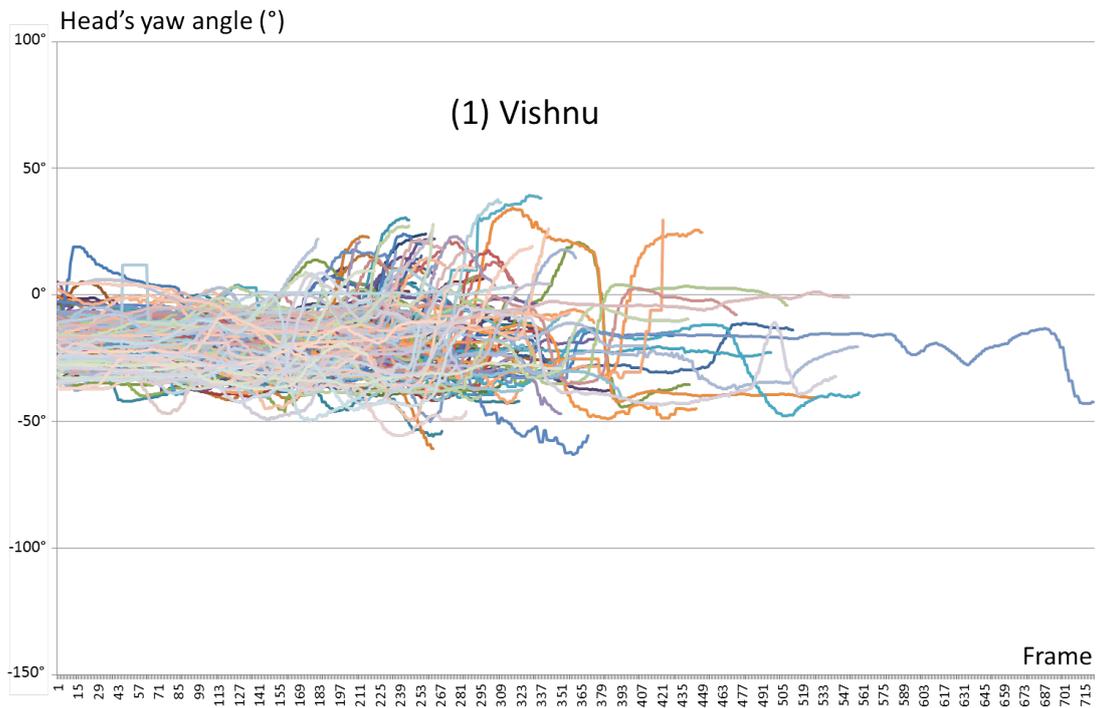
3.5.2.2 Discussion

We validated our hypothesis H2. In particular, qualitative results are very encouraging concerning our technique (mode 1), except for comfort (visual and global). This is probably due to the use of the video see-through HMD that slightly alters the perception of the real environment and needs a learning process in order to match visual feedback and proprioception. Moreover, other HMD limitations, such as the screen resolution or the latency, as well as perceptual issues, such as the conflict between the convergence and accommodation distances, also contribute to a downgraded user experience compared to mode 2 that does not alter perceptual abilities of subjects. Thus, it could also explain why we did not validate H1 in a pretty simple task context (L1): there were only 13 targets by sides (26 in total, cf Fig. 3.20b).

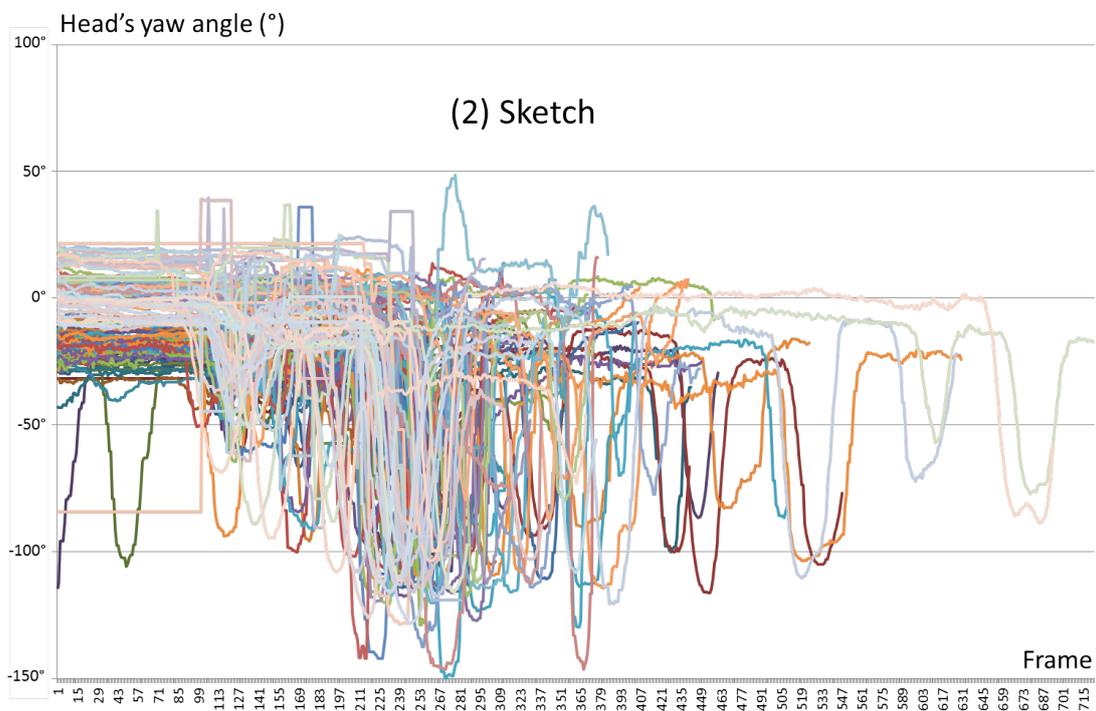
Next section presents results in a more complex task (L2), i.e with a bigger set of more dense targets with uniform shapes. In this context, mode 1 could be significantly faster than mode 2 because the mapping complexity would override the perception issues generated by the use of the video see-through HMD in mode 1.

3.5.2.3 Second targets' layout

For L2, we proposed targets amongst 78 uniform squares arranged in a regular grid (cf Fig. 3.20c). The idea was to remove spatial cues in the subject's real workspace in order to increase the mapping complexity of the correct targets in mode 2. It is similar to real world



(a)



(b)

Figure 3.24: Head's yaw angle for all iterations of all subjects. It illustrates the advantage of the fully collocated setup (mode 1) regarding the smaller range, 103° , versus 195° for the other mode (2). In sketch mode (2), we can identify goings and comings of the subject's head between the desktop screen and the real targets' space.

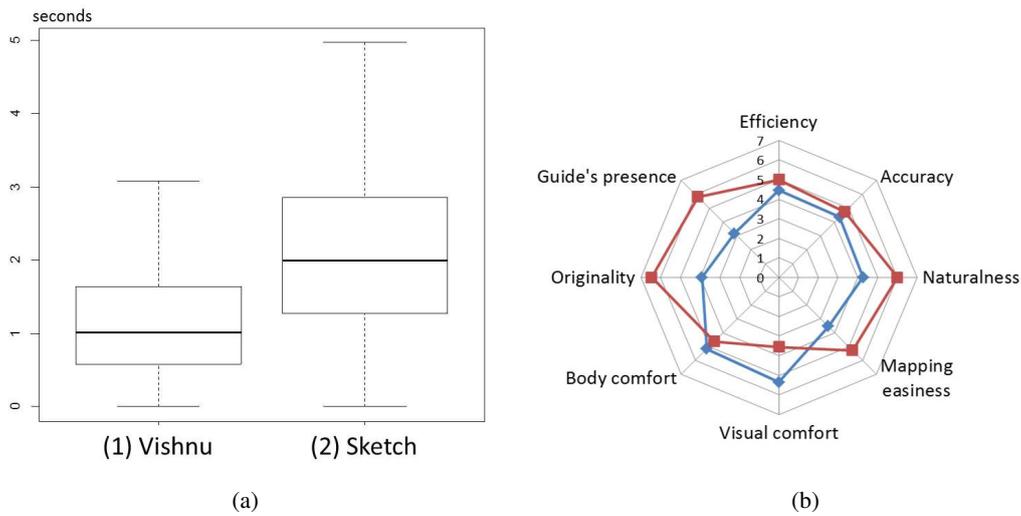


Figure 3.25: Results for the second layout L2. (a) Boxplot of the difference between the guidance completion and the subject's task. (b) Synthesis of qualitative results.

equipments such as computer racks or electrical panels that contain a lot of outlets arranged in a regular grid. Figure 3.25a shows task completion boxplot. In this more complex configuration, mode 1 is significantly faster than mode 2: $F(1, 432) = 41.56, p < 10^{-9}$. Concerning qualitative results (cf Fig. 3.25b), they remain the same compared with the first layout L1.



Figure 3.26: (a) The stretchable guiding arms, controlled by the remote expert’s location, from the agent’s viewpoint. On the left, the arms are extended and red because the guide is ahead of the agent. On the right, the agent has reached the guide’s location and the guiding arms retrieve their initial length and green color. (c) A simple guiding technique based on directional arrows and viewing frustum of the remote expert.

3.5.3 Navigation guiding

We just proposed a novel interaction technique based on a MR system in which a remote expert using a Virtual Reality (VR) application guides an agent using an Augmented Reality (AR) application. The agent must perform a physical task that is guided by virtual arms collocated with his shoulders, controlled by the remote expert. In our previous pilot study, we simplified our scenario by seating the operator. In more realistic cases, this is more complex. Indeed, before performing the actual arm guidance, the remote expert must reach the correct location in the virtual world while showing it to the agent in his augmented real world. To perform this initial location guidance, we propose to extend the GoGo technique initially developed for single user selection and manipulation tasks in VR [Poupyrev et al., 1996]. We added the arms representation as in [Fraser et al., 1999] and removed the non-linear scaling, instead controlling it directly with the expert’s interactions.

3.5.3.1 The stretchable guiding arms

The stretchable guiding arms technique has been developed to overcome the limitations found in the techniques from the literature in a specific context: the expert and the agent share a fully collocated setup. Moreover, the final goal of the system is to guide the agent in performing a physical task with the use of virtual arms controlled by the remote expert. In this setting, the first step consists in viewpoints collocation. Then, the expert must show which object to move, and thus he often needs to move the agent’s viewpoint forward in order to grasp the virtual cloned object. The existing technique based on directional arrows can achieve this purpose, as illustrated in Figure 3.26b, but first users who tested our system complained about eye strain because of the proximity of the visual cues, including the remote expert’s frustum. Moreover, it increases the cognitive load of the agent who has to follow arms guidance and location guidance at the same time using different visual cues. To improve this, we take advantage of the already existing virtual arms initially present for gesture-based guiding purposes only. They can also be used for location reaching guidance in a non-intrusive way, contrary to other methods. Figure 3.27 illustrates the principle of the proposed stretchable guiding arms technique. When

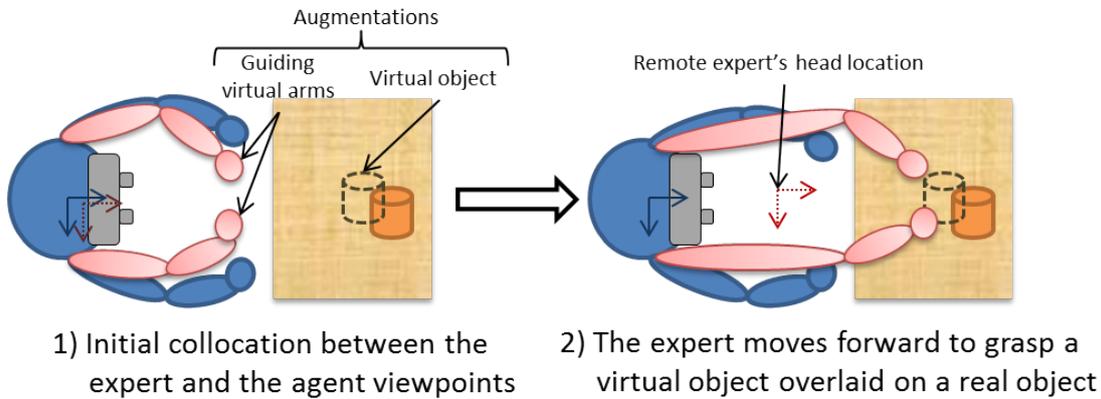


Figure 3.27: Principle of the stretchable guiding arms technique. Illustration from the agent side in blue. From the expert's viewpoint (in red), his virtual arms do not scale and also stay collocated with his shoulders.

the expert chooses to switch to stretch mode, his forward motion makes the virtual arms seen from the agent's viewpoint start to scale up to keep the virtual hands always collocated with the expert's ones, while maintaining the virtual shoulders fixed according to the agent's shoulders. From the expert's viewpoint, his virtual arms do not scale up, and also stay collocated with his shoulders.

The main advantage of this technique is the naturalness of the guidance. Indeed, the agent can directly perceive the forward motion of the remote expert, thanks to the scaling up of the virtual arms. Then, to follow the expert, the agent simply moves his/her viewpoint in order to retrieve the initial length of the virtual arms. In addition to the scaling cue, a coloring code smoothly changes the color of the independent virtual arms from green to red, according to their scale. They return to green when the scale is close to its initial value, which means that the agent has reached the expert's viewpoint (cf. Figure 3.26a). If the agent overtakes the expert's location, the virtual arms keep their initial length and color, but their bending indicates to the agent that he must move backward to retrieve a correct pose.

In parallel, the expert can use a virtual camera with a different viewpoint of the VE in order to get visual feedback on the agent following. This camera is displayed in a corner of the expert's viewport and is enabled when the stretchable guiding arms technique is activated. The remote expert can switch between different available viewpoints: a side or front view based on his/her head location, or the viewpoint of the agent slightly behind him when performing a forward motion guidance. Moreover, the color change for virtual arms from green to red is used to inform the remote expert explicitly about his/her distance with the guided agent.

3.5.3.2 Pilot User Study

We compared reaching performances and subjective feelings of (C1) our technique (cf Figure 3.26a) with (C2) the frustum with 3D directional arrows based approach (cf. Figure 3.26b). In order to evaluate the agent's performance in the location reaching process, we try to remove bias generated by the expert's interactions. Thus, we recorded the interactions of a real expert in a pre-process phase in order to replay it for the experiment, removing possible expert's interaction bias due to changing behavior. We postulate that the expert's interactions are not really

affected by the agent's ones. Indeed, the expert must reach a location to grab a virtual object, and thus must move forward to achieve the goal assuming that the agent is following.

Measurements and Hypotheses We collected three objective measures: completion time (hit target around 0.15m), mean distance between the replayed expert and the subject, and final precision (triggered by the subject) in location and orientation. Moreover, subjects fill in a final subjective questionnaire (7 point Likert-scale) to express their impressions of the naturalness of the system, their perceived cognitive load and eye-strain, amongst other considerations. We stated the following hypotheses:

- H1: Completion time, mean distance and precision are equivalent in both conditions C1 and C2
- H2: C1 is more comfortable to use than C2

Protocol We design our experiment in a counter-balanced way using a set of 20 pre-defined targets (10 per condition) in a range from 1 to 3 meters from the home location and a horizontal angle range from -30 to 30 degrees (cf. Fig.3.28). Their altitude is defined around a mean value of 1.5m with a range of ± 0.3 m. Each participant passes both conditions twice alternatively, with a different set of 2x5 virtual targets per condition. These targets are not displayed for the replay process because the subjects must reach them following only the recorded guide. We simulate the initial viewpoints collocation, illustrated in step 1 of the process described in Figure 3.27, with a re-targeting of the replayed data according to the subject's height and exact home location.

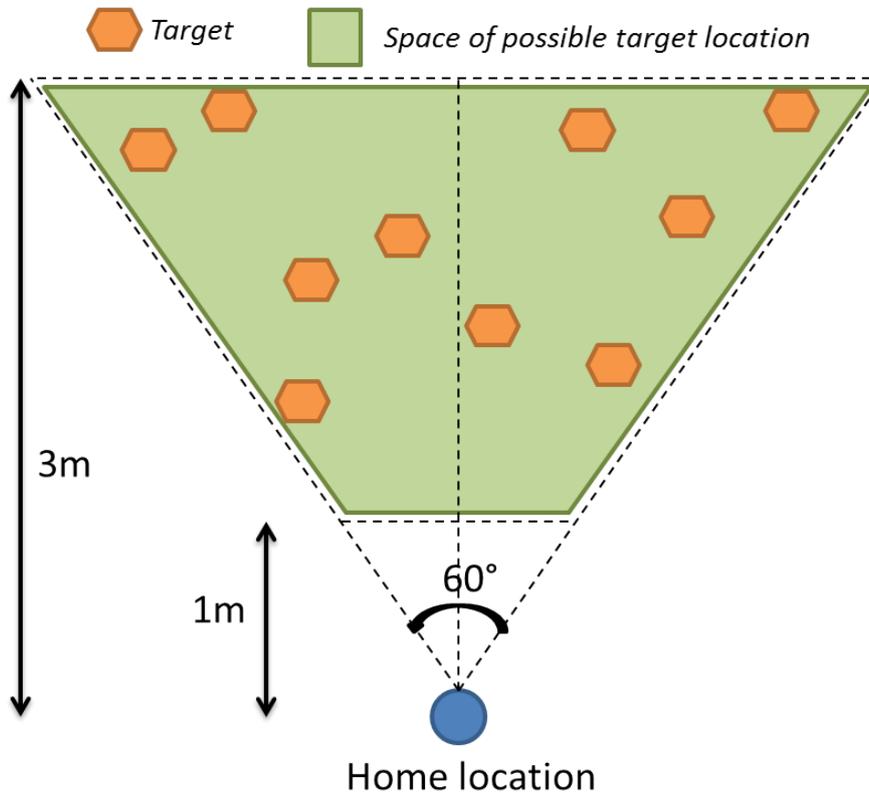


Figure 3.28: Example of 10 targets defined for one condition.

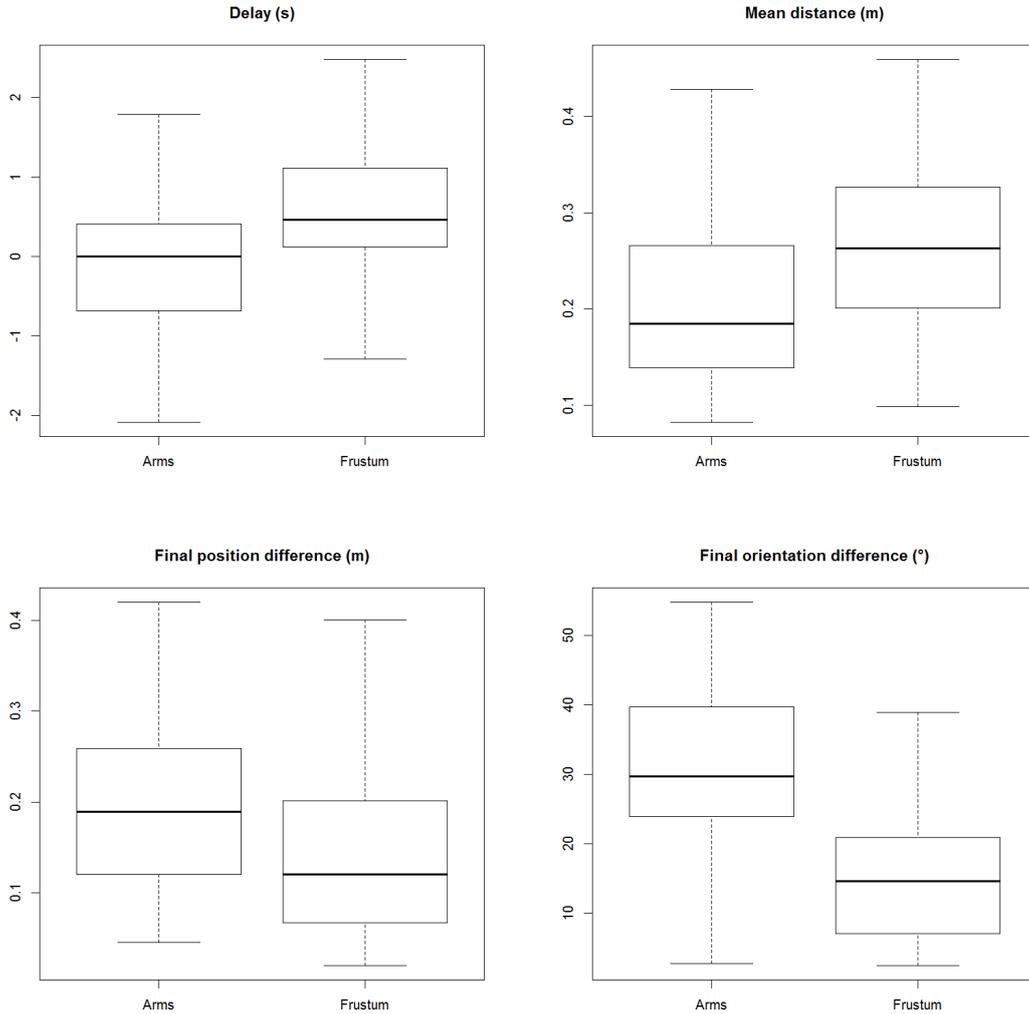


Figure 3.29: Boxplots of the delay between the target hitting by the guide and the subject, their mean distance until the hit, and their final position and orientation differences.

Results Results have been collected from 10 subjects aged from 20 to 30 ($mean = 24.4, sd = 4.03$). A few iterations are not part of the analysis because of tracking issues. Figure 3.29 illustrates the collected quantitative results. We found a significant difference in the target hitting delay between the guide and the subject ($F(1, 194) = 20.47, p < .01$). This delay is shorter with (C1) the guiding arms ($M = -0.03, sd = 1.28$) than with (C2) the frustum ($M = 0.75, sd = 1.61$). In the same way, the mean distance between the guide and the subject until the hit is significantly smaller ($F(1, 194) = 20.39, p < .01$) using (C1) the guiding arms ($M = 0.21, sd = 0.008$) than (C2) the frustum ($M = 0.26, sd = 0.007$). In terms of precision, the frustum-based guiding (C2) is significantly more accurate than the guiding arms (C1) for both location ($F(1, 194) = 11.34, p < .01$; (C1): $M = 0.19, sd = 0.006$; (C2): $M = 0.15, sd = 0.01$) and orientation ($F(1, 194) = 55.37, p < .01$; (C1): $M = 31.10, sd = 122.55$; (C2): $M = 16.54, sd = 252.88$).

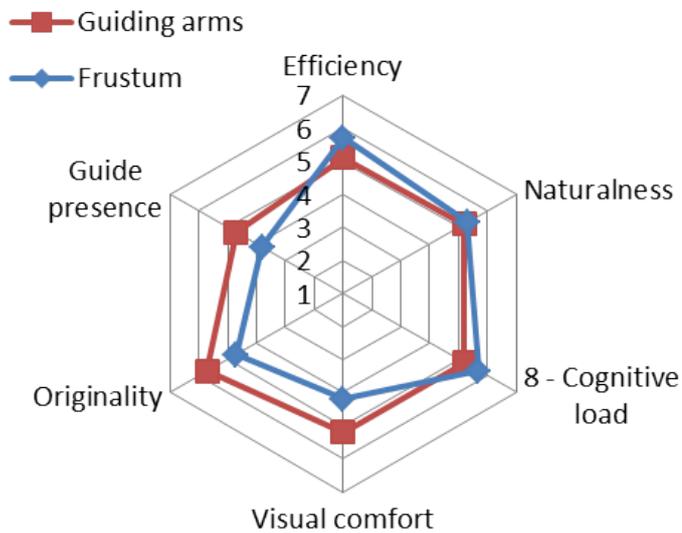


Figure 3.30: Synthesis of subjects' qualitative appreciations.

Discussion H1 is not proved. Actually, we found that the guiding arms (C1) are better for closely following the remote guide, while the frustum-based guiding (C2) is more accurate for final precision. This degradation of accuracy using C1 can be explained by the complexity of human body's DOFs that generate variations between the relative pose of the agent's head and the expert's hands. Thus, the coupling of C1 with an alternative guiding technique for final positioning should be studied to improve accuracy without decreasing naturalness of the guidance. Concerning the qualitative results (cf. Figure 3.30), we validated H2 because of better visual comfort using our approach (C1). Moreover, subjects prefer our approach (C1) in terms of originality and guide presence. For other criterion (efficiency, naturalness and cognitive load), results are similar for both conditions C1 and C2.

3.5.3.3 Extension

As results are better for C2 in terms of final precision for position and orientation, we introduce an additional guiding cue with the C1 technique. This extension is enabled as soon as the agent is close enough the guide and handles both position and orientation guidance. Thus, it provides a final 6 DoF guiding cue in order to improve final positioning. This extension is based on an alteration of the final rendering in order to guide the agent in the right looking direction taking also into account its position. Figure 3.31a illustrates the way it computes the pixels to alter according to both users position and orientation. It is implemented using CG fragment shader and a reticule texture (cf. Figure 3.31b) used to alter pixels. Figure 3.31c illustrates its action on the agent's field of view.

This additional guiding cue must be formally evaluated, but according to first tests, we think this is a good way to improve our stretchable guiding arms technique for final precision. It generates a visual discomfort in the follower's field of view, that tends to reorient the follower toward the guide's point of view. In the same time, its implementation is not so disturbing as it only acts on the peripheral vision area of the user; while avoiding the use of the guide's frustum that generates uncomfortable virtual inter-penetration between the follower's point of view and

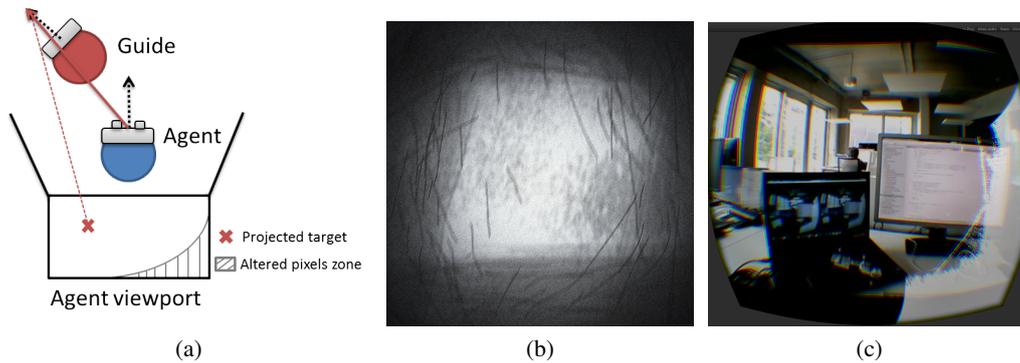


Figure 3.31: (a) Principle of the guiding extension to improve final precision, from top view. (b) Reticule noisy texture used to alter pixels within the guiding shader. (c) Illustration of the guiding shader performing as an extension of the stretchable guiding arms. The white zone is the alteration zone meaning that the guide is looking to the left and must be a little higher than the current follower's viewpoint.

the guide's virtual frustum.

3.6 Conclusion

In this chapter, we developed ideas around the use of MR technologies to improve remote collaboration. Especially, we applied this to maintenance in order to let a remote expert help an agent in performing a procedure. We distinguish two phases in this scenario. First, the remote expert must be able to analyze the agent's workspace in order to diagnosis the procedure to perform. In the same time, he must guide the operator to place him in the correct location. Thus, we proposed guiding technique for remote navigation. Second, the physical manipulation can begin. For this phase, we proposed guiding techniques for selection and manipulation tasks. In this context, we investigated two kind of systems. The first asymmetrical one is inspired from existing systems based on HWD with remote camera streaming. We improved it using 3D capturing of the operator's workspace in order to provide an enhanced AR view for the operator as well as a better perception, including depth, for the expert. It allows 3D interactions for the expert that can use 3D visual guiding cues, perfectly integrated in the operator's real world, in order to help him in achieving the procedure. The second symmetric approach aims to free the hands of the operator, while improving expert's interactions capabilities and easing operator's perception of the guiding visual cues. Here, the remote expert is immersed into the operator's workspace at 1:1 scale. We provide to the expert two virtual arms in order to interact with virtual objects. The main idea of our proposal is to collocate the expert with the operator's point of view. Then, the virtual arms, also coming out the operator's shoulders, are used as natural guiding cues that show what are objects to grab and how to move them.

We ran several pilot user studies in order to evaluate the efficiency and the acceptance of our prototypes. Results show good performances using our systems compared to already existing ones. For navigation, especially in building containing corridors and complex room shapes, our approaches add continuity between the operator and the pointed spot in order to always keep him aware of the expert's guidance. In our symmetrical fully collocated setting, we found some drawbacks about precision of the guidance because of a less intrusive but coarser visual guiding

cue. Thus we proposed an extension in order to improve this aspect that should be evaluated in future work. Concerning selection and manipulation to actually perform the physical manipulation in order to achieve the maintenance procedure, our preliminary user evaluation shows that participants perform better using our fully collocated setting when dealing with complex layouts similar to computer racks or electrical panels. However, further experiments should be run in order to formally evaluate the performances of our both settings in realistic scenarios.

Perspectives In this work, we separated our two proposals into two different systems: symmetrical and asymmetrical. Our findings show that according to the situation and the phase of the procedure, the expert may need to freely navigate into the remote operator's workspace, potentially at different scale, or may need to share the operator's point of view to better understand his situation. Thus, we plan to merge both settings into an unified prototype that would handle these different situations. We could even imagine a system that would combine smartly both displays, i.e hand-worn and head-worn, with the possibility for the operator to switch from one to another, or even to use both at the same time in order to provide additional capabilities for the expert to perform the guidance.

Chapter 4

Co-manipulation

In the context of remote collaboration, VR allows to share purely virtual world while being physically distant. In this virtual world, remote collaborators can accomplish tasks that simulate real world task in a local context. An interesting challenge is to provide some ways to manipulate an object collaboratively, which is a common task in the real world. Thus, after the focus on our use case that is remote maintenance using MR, we contribute to the field of co-manipulation in VR that can be used to train with reproductivity capabilities, to learn about procedures and also to accustom to work with unknown people. Moreover, compared with the previous chapter 3, the co-manipulation task can imply more than two users interacting in a shared VE and can also propose more asymmetry in terms of collaborators' roles.

First, we present a brief state of the art of co-manipulation in section 4.1. Then, section 4.2 proposes a generic approach that we have implemented to co-manipulate virtual objects. The main feature of our approach is its capacity to handle heterogeneous interaction devices in a generic way. Last, section 4.3 goes further in virtual co-manipulation concept by proposing an hybrid approach that integrates the manipulation of a user's PoV by another user. This is close from the remote guiding techniques that have been investigated in the previous chapter.

4.1 State of the Art

In this work, referring to the classification proposed by Margery et al. [Margery et al., 1999], we focus on cooperation at level 3. We want that the different users can change the same object transformation at the same time. This kind of synchronous collaborative object manipulation in CVE can be divided in two families: symmetrical and asymmetrical approaches. In the first symmetrical approach, collaborators benefit from equivalent viewpoints and interaction capabilities. This collaboration can be homogeneous based on 3D ray [Pinho et al., 2008] or 3D pointer (i.e virtual hands metaphor) [Duval et al., 2006a, Aguerreche et al., 2009b], or heterogeneous, for instance mixing a 3D ray for a user, and a 3D pointer for another one [Pinho et al., 2002]. On the contrary, in asymmetrical approaches, the users collaborate in a multi-scale CVE, i.e they have different viewpoint scale, thus their interaction techniques must be adapted to their different capabilities [Duval and Fleury, 2009, Fleury et al., 2010a].

Overcoming the loss of the physical link The main issue introduced with virtual co-manipulation is the lack of a physical link that can greatly decrease its effectiveness. Indeed, in real life co-manipulation, users are carrying the same physical object that creates a tangible link between

them. This is essential to synchronize their motion and complete the manipulation task. In VR, and even more in remote context, this physical link is not available. Thus, techniques must be proposed to substitute this loss. In local context, we can use a tangible prop as in [Aguerreche et al., 2010] that proposes a reconfigurable tangible device able to be reshaped in order to match the global shape of the manipulated virtual object. In remote context, haptic devices can be used to simulate reactions generated by the co-manipulation task [Sallnäs et al., 2000]. Thus, the handle can pull, push or twist the user's hand according to the merge of his and others inputs. In this case, an average method is used to manipulate the object that mimics real world cooperative manipulation.

Other less intrusive approaches propose vibro-tactile feedback [Naud et al., 2009]. The advantage is that users do not have to use a cumbersome haptic arm that has a limited motion space. Last, visual feedbacks can also be used to simulate haptic feedbacks [Sreng et al., 2006]. It is called pseudo-haptic feedback and is based on the dominance of visual sensory over the proprioceptive one. However, these approaches can not provide real haptic sensations and can not physically restrain the motion of users, and thus they must propose adapted interaction metaphors.

4.2 Generic Aggregation-based with Heterogeneous Devices

Many interaction devices can be used in VR applications and even more are put on the market regularly. Moreover, interaction metaphors must be adapted according to the devices capabilities. As an example, users can have 6DoFs controllers in their hands that take advantage of natural hand skills. An associated interaction metaphor can be based on a 3D pointer or a 3D ray. If not available, a 3D ray could be controlled through a gaze-based metaphor. This is a simple example that illustrates the potential complexity to deal with when developing interaction techniques for collaborative VR applications due to asymmetrical devices setup.

Our approach aims at easing the integration of different devices and new interaction techniques into a generic framework handling co-manipulation in a potentially multi-scale CVE (that can introduce different roles for each collaborator in the co-manipulation task). Figure 4.1 illustrates the asymmetrical dimensions managed by our approach. The co-manipulation could be based on level 3.1, splitting the DoFs between users, but we rather implemented level 3.2 giving the ability to concurrently modify the all 6DoFs of the manipulated object at the same time (cf. section 1.2.4). However, downgrading to co-manipulation at level 3.1 is simple to achieve because the model already manages level 3.2. To provide such a framework, we propose a solution based on physically-simulated manipulation through virtual handles and connected springs. Similar approaches have already been proposed [Koutek and Post, 2001, Fröhlich et al., 2000], but they either do not address collaborative manipulation, or do not handle heterogeneous devices. Other work even uses a spring-based manipulation technique to provide natural whole-hand interactions while handling the visual inter-penetration issue [Borst and Indugula, 2006]. Indeed, spring-based manipulation is an interesting approach to manage multiple inputs in a generic way and with scalability capability based on physical simulation.

4.2.1 Model description

In order to provide a generic way to co-manipulate a virtual object, we propose to perform this manipulation through virtual handles initially stuck to the object's surface. Using physical shapes that match the visual shapes of virtual objects, this approach can bring grabbable capa-

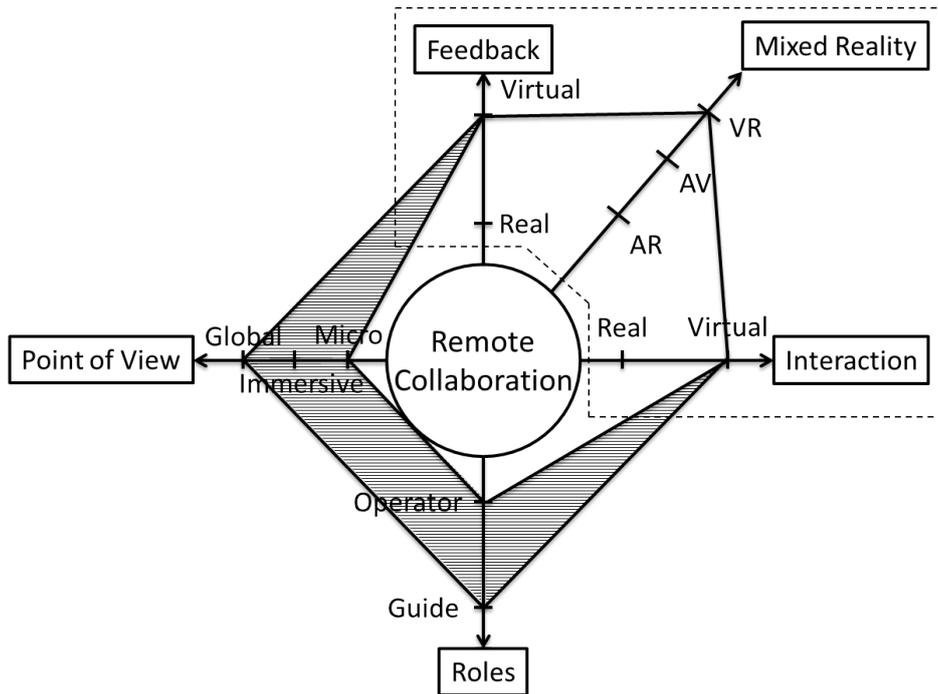


Figure 4.1: Asymmetrical dimensions available with our generic approach for co-manipulation. The gray hatchings mean that users are not required to use pre-established configurations but can choose amongst these all combinations.

bility to any object in a simple way. Then, the interaction metaphor must handle how the user can select a spot on the object surface to dynamically stick a virtual handle on it. Actually, to be integrated in our model, a new interaction technique must only manage how to update the position of the used interaction tool and how to trigger the grab and release of a handle. The remaining of the needed behavior for co-manipulation is managed in a generic way. For instance, when trying to grab a handle, the framework checks if the interaction tool intersects a surface object or an already created handle. In the first case, it creates a new handle on the object surface and grabs it, and in the second case it just grabs the existing handle. Once a handle is grabbed, a virtual spring is created between the virtual handle and the interaction tool. If only one handle manipulates the object, the manipulation is direct because the handle and the interactive tool are merged (the spring length is null). As soon as more than one handle is stuck on the same object, the interaction tool is still updated the same way, but the different constraints applied on the object create a difference between the interaction tool position and its grabbed handle. This difference is displayed in order to add a visual continuity using a virtual spring. Moreover, the co-manipulation is computed with a mass-spring model, thus the visual representation of these springs helps the user to understand what happens adding semantic. Last, these springs are useful to stay aware about what others are doing in the shared VE and to improve the understanding of the object's behavior while it is manipulated by several users at the same time.

In order to stabilize the manipulation, we used a pretty high damping factor for the springs. Moreover, to manage direct manipulation with only one handle, the stiffness of the spring is also very high to keep its length null in this specific case. Finally, our approach implements an average method for co-manipulation using physical simulation to compute the final result.

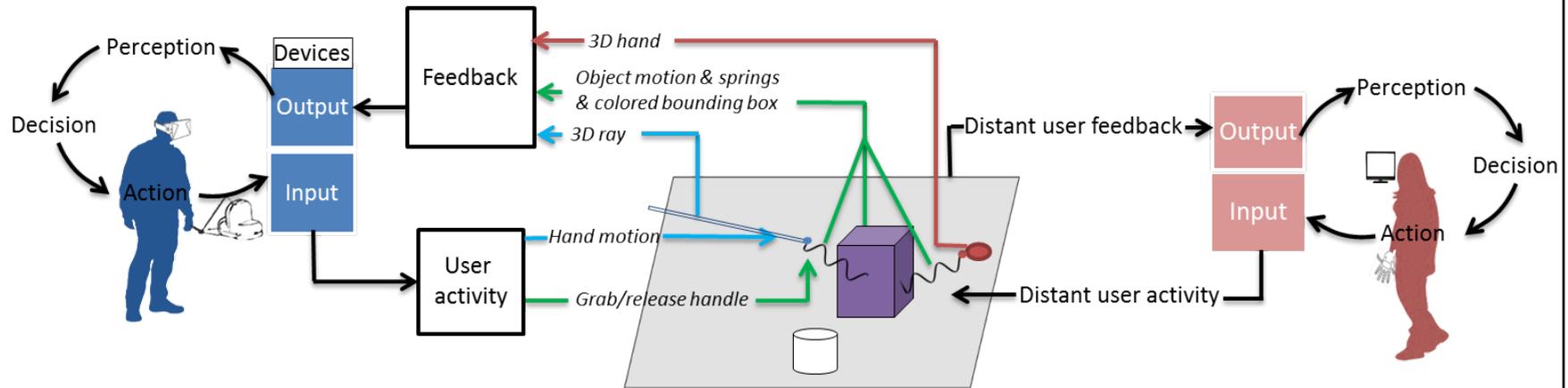


Figure 4.2: Description of our generic approach for co-manipulation instantiating our awareness model.

4.2.2 Visual feedbacks

Figure 4.2 describes our approach according to our awareness model for CVE and Figure 4.3 illustrates the manipulation of a virtual object by one to three users at the same time. On the top screenshot, the yellow user manipulates alone the object with a direct mapping between his/her interaction tool (the yellow 3D pointer) and the virtual handle, thus the spring is not visible. A transparent bounding box is added around the object with the color of the current owner. On the middle screenshot, the second red user adds a handle on the object. His/her interaction tool is a 3D ray that can pick handles and move them. Now, the springs are visible because the two users concurrently manipulate the object. The bounding box color goes orange that is the average color of the two involved users. Last, on the bottom screenshot, a third green user adds another handle and picks it with a 3D pointer. When an object is manipulated by more than 2 users, the bounding box color goes white.

4.2.3 Asymmetrical devices and PoVs

Interaction techniques based on many devices have already been integrated in our framework. The interaction metaphors are based on a 3D pointer (3D hand) or a 3D ray. In this second case, the position of the interaction tool is defined at the end of the 3D ray that continuously uses raycasting on objects of the scene in order to define its length as the distance from the beginning of the ray and the first intersection point. Some interactions are uni-manual, others are bimanual and all of them allow collaboration with potential different PoVs on the scene, enabling multi-scale collaboration. We can notice that an implementation of a complex virtual hand with grab capability for each fingertips could also be integrated to provide whole hand interaction into the VE.

Many devices have been integrated in our framework thanks to the provided model that ease this process:

1. Leap motion
2. Kinect
3. Razer hydra
4. zSpace (with 3D stylus)
5. HMD (based on head orientation)

(1) and (2) can either control a 3D pointer, or a 3D ray; the `grab()` function is called when the hand closes and the `release()` function is called when it opens. (3) and (4) are 3D trackers with buttons. The same way they can control a 3D pointer or a 3D ray, and the functions are called according to a button state (triggered or not). Last, HMDs provide rotational head-tracking that can be used to define a 3D ray based on head rotation. The grab and release of handles is implemented using physical buttons of the keyboard or the mouse, but it could also be controlled with a button integrated on the headset.

4.2.4 Improvement tracks

Our proposal provides a generic co-manipulation framework to integrate new interaction devices, as well as new interaction metaphors, in an efficient way. However, in the current implementation, only 6DoFs of the object transform can be modified: its position and rotation. The scale is not handled because it introduces an additional transformation mode that is difficult to manage without an explicit application control. Nevertheless, it would be easy to adapt the

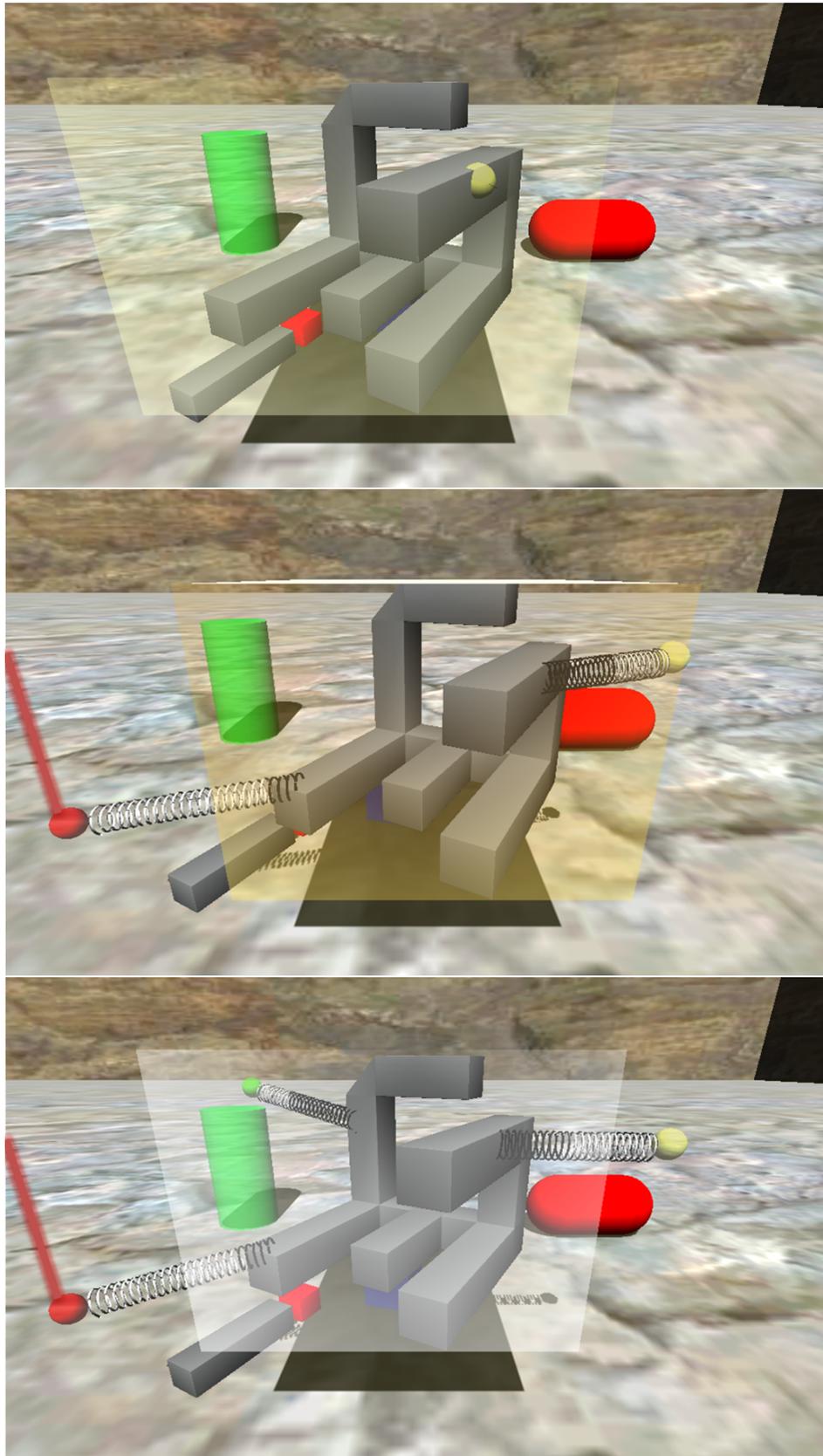


Figure 4.3: Illustration of a virtual object manipulated by one user, and then concurrently manipulated by two and three users at the same time.

framework in order to manage the all 9DoFs of a virtual object, explicitly switching between a manipulation mode and a scaling mode.

Another refinement could be to associate each user with a priority factor depending on its role in the co-manipulation process. For instance, if a user is the instructor, the system could give him/her a higher priority in the co-manipulation in order to let him/her lead the process. In the context of training other people, it is a great feature to better help them while still collaboratively manipulate a virtual object. The implementation of such a feature into our framework is pretty trivial since it just needs to multiply the stiffness factor of the virtual springs with the user-specific priority factor in order to give it more importance in the co-manipulation process.

4.3 Multi-Scale Hybrid Split of DoFs

In the previous section, we have proposed a generic framework to manage co-manipulation with asymmetric devices and PoVs. From this base approach, we aim to address a specific case of co-manipulation close from a practical use that we have investigated in our use case that was remote guiding in chapter 3. Here, we merge remote guiding and co-manipulation in a pure virtual environment by developing a hybrid technique that involves to move the PoV of another user while still collaboratively manipulating the same object.

To do so, we propose a set of metaphors that enables several users to collaborate in a shared multi-scale VE [Ibayashi et al., 2015, Nguyen et al., 2013] in order to achieve collaborative manipulation tasks in VR. Such tasks could consist in overcoming different obstacles by moving, rotating and scaling an object collaboratively. This kind of collaborative manipulation can be used to simulate industrial tasks such as in an automotive factory where cumbersome objects must be carried by several collaborators [Aguerreche et al., 2009b]. Thus, we propose an asymmetric collaboration between two or more users with different devices (cf. Fig. 4.5). These users are embedded in a co-located multi-scale VE thanks to a model inspired from the Immersive Interactive Virtual Cabin (IIVC) generic model [Fleury et al., 2010a].

Our system provides an efficient way to co-manipulate an object within irregular and narrow courses, taking advantages of asymmetric roles in synchronous collaboration. Moreover, our technique has been developed in the context of the IEEE 3DUI Contest 2016. Thus, in order to fulfill the contest requirements, our technique provides a way to maximize the filling of the courses while the object moves on its path.

Moreover, one of our goals is to benefit from the wide variety of today VR devices. Therefore, our solution is based on an asymmetric collaboration pattern at different scales in which users benefit from suited points of views and interaction techniques according to their device setups. Figure 4.4 illustrates the asymmetrical dimensions introduced in our approach.

4.3.1 Motivations

Our approach is motivated by the exploration of innovative ways to manage co-manipulation that can introduce some awareness issues. Indeed, the main idea is to put a user into the manipulated object, with a micro PoV into the shared VE, while another user has a global PoV of the shared VE. Both users are able to co-manipulate the same object at the same time. By placing one user inside the manipulated object, we introduce a hybrid approach for co-manipulation that requires adequate feedbacks in order to let users being able to maintain a correct awareness of their activity, activities of others and the state of the manipulated object.

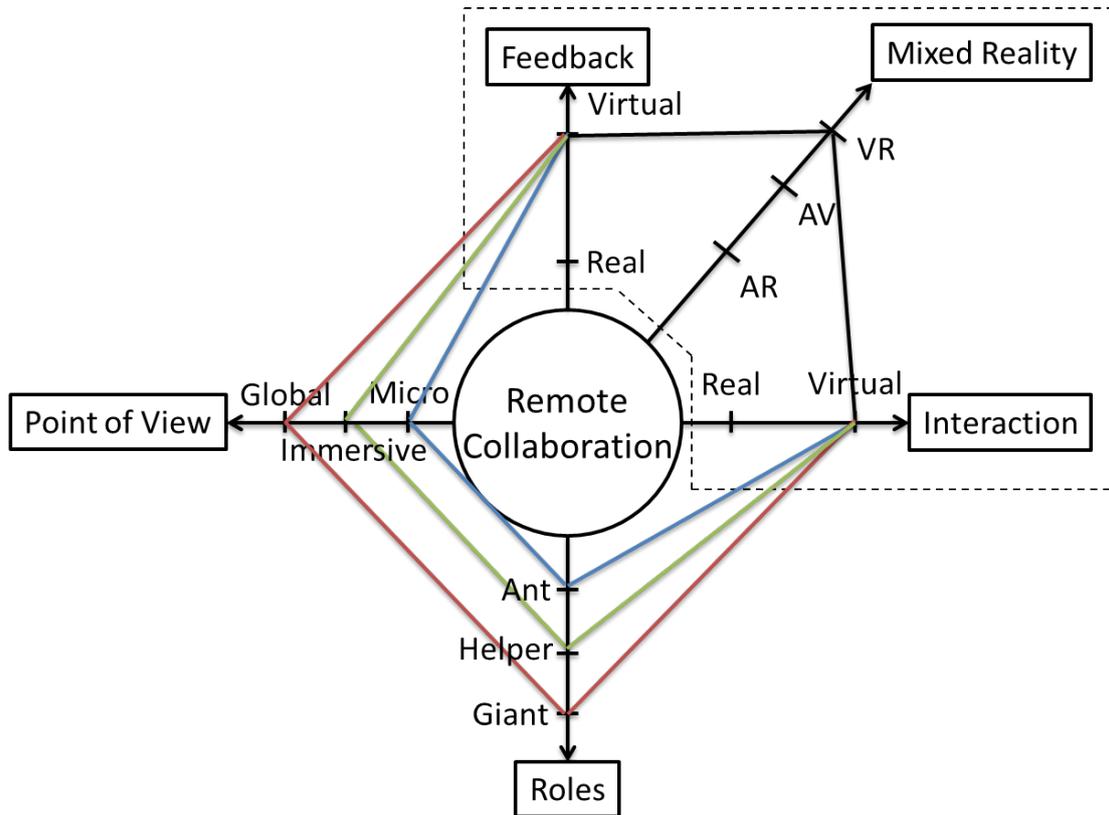


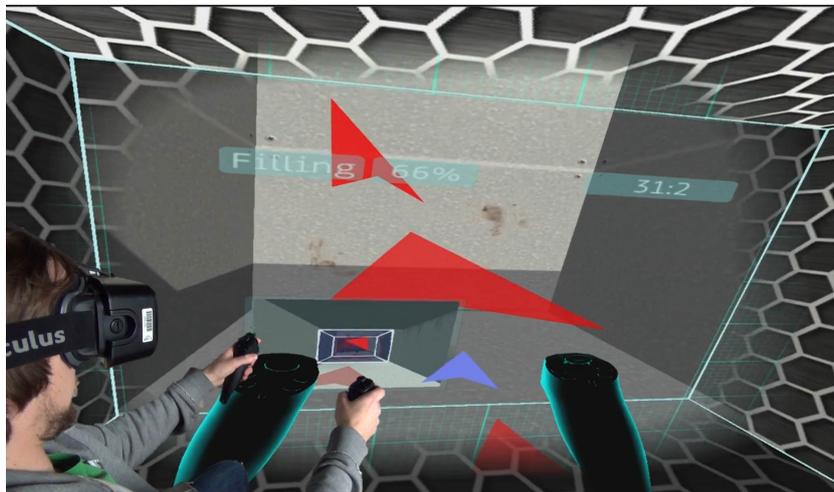
Figure 4.4: Asymmetrical dimensions introduced in our hybrid approach for co-manipulation.

Moreover, this work takes advantage of the current state of the VR market. All devices used are already available on the consumer market: a zSpace, an Oculus Rift with a Razer Hydra, a Gear VR and a Google Cardboard. Indeed, today there is growing number of available VR visualization and interaction devices. Each setup offers different interaction capabilities. That is why our approach benefits from this diversity by proposing a set of interaction techniques adapted to different device setups in order to perform a collaborative manipulation task. These techniques are automatically associated to the suited users with plasticity mechanisms. Plasticity is defined as the capacity of an interactive system to withstand variations of both the system physical characteristics and the environment while preserving its usability [Thevenin and Coutaz, 1999]. Here, we demonstrate an example of plasticity for device and collaboration adaptations thanks to the models presented by Lacoche et al. [Lacoche et al., 2015a].

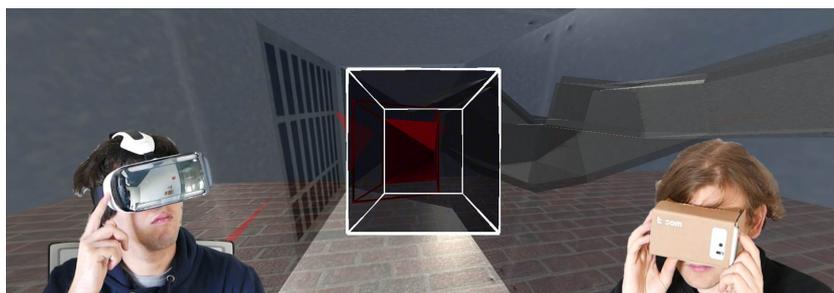
As proposed by Pihno et al. [Pihno et al., 2002], our approach splits the Degrees of Freedom (DoFs) of the manipulated object between collaborators in order to maximize the system efficiency: the *Giant* (with a global viewpoint) controls the object's translation, while the *Ant* (inside the object) sets its scale and rotation. In our shared multi-scale VE, this approach allows the *Giant* to quickly move the object, while the *Ant* performs better accurate transformations and has a suited viewpoint to maximize the courses filling. The object's rotation can also be shared between the two main users using a particular fusing scheme inspired from the asymmetric integration by Ruddle et al. [Ruddle et al., 2002]. In this case, we use a non-linear merging factor for the *Giant*'s action according to its ray bending. Last, the third helping user (with the 3rd person viewpoint on the object) can also set the scale of the manipulated object by asymmet-



(a)



(b)



(c)

Figure 4.5: Collaborative manipulation of a virtual object (here, a cube) based on an asymmetric setting between two users who can be helped by two additional users. (a) The first participant, called the *Giant*, has a global view of the scene and moves the object with a 3D bent ray. (b) The second user, called the *Ant*, is placed inside the object and precisely rotates and scales it. (c) Two additional roles can be added. The first one helps to scale the object using a third person view of it. The other one is a spectator who switches between the other participants' viewpoints and helps them with oral communication.

rically integrating its action with the *Ant*'s one. All metaphors for controlling these DoFs have been designed in an iterative way in order to optimize the learning curve for the end user.

In order to demonstrate the interest and the efficiency of our solution, we have proposed to test it in the context of the IEEE 3DUI contest 2016¹. The goal of this contest was to move a cube collaboratively through different labyrinths, while maximizing the courses filling of the object. Most of our illustrative examples are based on the manipulation tasks required by this contest.

First, an overview of this work is presented in Section 4.3.2. Second, we present collaborators' roles of our proposal in Section 4.3.3, then we detail interaction techniques in Section 4.3.4. Section 4.3.5 describes our implementation and Section 4.3.6 discusses informal feedbacks of test users. Last, we conclude and present perspectives of this work in Section 4.3.7.

4.3.2 Overview

Co-manipulation techniques can be implemented in two different ways: split of DoFs (level 3.1) or concurrent modification involving a merge policy (level 3.2). In our approach, we use both conditions in order to maximize the efficiency of the whole proposed approach. Basically, the object manipulation uses a split of DoFs policy between the *Giant* and the *Ant*. However, in some conditions, the *Giant* can take the lead on the *Ant* rotation capability in order to help him. This is done by merging both inputs with a custom policy giving a bigger influence to the *Giant* action. Then, if a third user joins the CVE, the scale of the object is controlled using a cooperative interaction with the *Ant* at level 3.2. We did not find any related work that uses this kind of adaptable approach. Moreover, our proposal involves a kind of hybrid manipulation/navigation technique for the *Ant*. Indeed, the *Ant* is placed inside the manipulated object, moved by the *Giant*. This feature involves the *Ant*'s navigation in the CVE, controlled by the *Giant*. However, the *Ant* does not control the manipulated object location, thus, the approach does not require any navigation technique even if the *Ant* may feel as in a car or a spaceship. This hybrid approach is also an innovative way to apprehend co-manipulation using an asymmetrical approach to improve the efficiency of this difficult task.

4.3.3 Asymmetric collaborative scenario

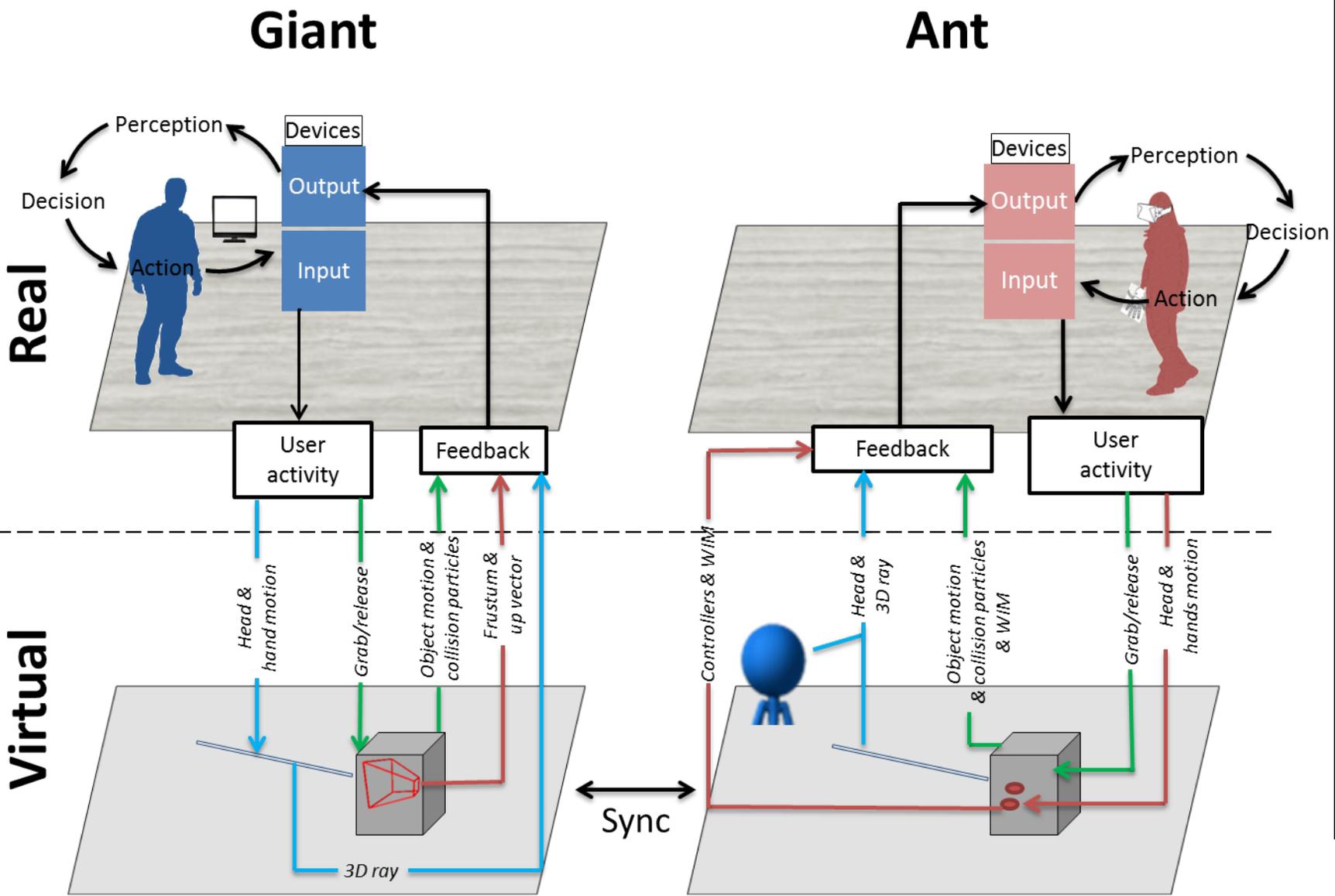
We propose an asymmetric collaboration where each user benefits from interaction capabilities adapted to his/her interaction devices in order to move, rotate and scale a virtual object. It is based on two main roles, a *Giant* with a global view of the shared environment, and an *Ant* inside the manipulated object. Figure 4.6 describes our approach using our awareness model. Two other roles are also possible for assisting the two main users.

4.3.3.1 Global View: the *Giant*

The first user is interacting on a zSpace² as shown in Figure 4.5a. It is composed of a 3D stereoscopic display with head tracking and of a 3D tracked stylus for interacting. The zSpace screen is used to create a window to the VE. Therefore, this user has a global view of the scene and can roughly manipulate the object in order to move it really fast in easy passages. This user can translate the object and also shares with the *Ant* the possibility to rotate it.

1. The illustrative video submitted for the contest is available online: <https://www.youtube.com/watch?v=Yp130VMMLQI>

2. <http://zspace.com/>



4.3. MULTI-SCALE HYBRID SPLIT OF DOFS

Figure 4.6: Description of our hybrid approach for asymmetric co-manipulation instantiating our awareness model.

4.3.3.2 Micro View: the *Ant*

The second user visualizes the scene with a Head-Mounted Display (HMD), here an Oculus Rift³ as shown in Figure 4.5b. The *Ant* is interacting with a Razer Hydra⁴ composed of two 3D tracked controllers. We exploit the immersion feeling given by the HMD to place this user inside the manipulated object. This position enables the *Ant* to manipulate the object with a fine accuracy. This role is essential to overcome difficult passages and maximize the courses filling by the object. The *Ant* can scale the object and shares the possibility to rotate it with the *Giant*. The *Ant*'s scale in the scene also offers direct interaction possibilities such as pushing buttons to trigger different actions.

4.3.3.3 Third Helping User

As shown in Figure 4.5c, the third user is interacting with a GearVR⁵, an HMD with a 2D trackpad. This role is optional. This user has a third person view of the manipulated object. His/her role is to help the *Ant* to scale the object with slide gestures on the GearVR trackpad. Therefore, the scaling capability is shared by these two users.

4.3.3.4 Spectators

The last role is a spectator. It is available on multiple devices. In our scenario, he uses a Google Cardboard⁶ as shown in Figure 4.5c. Multiple spectators can be included in the shared VE. These users can switch between the other participants' viewpoints. Here, it is done by pulling the Cardboard trigger. They can help the other participants by giving oral instructions.

4.3.4 Interaction Techniques

The collaborators benefit from complementary interaction techniques to perform collaborative manipulations task that need translating, rotating, and scaling. An example of task is given in Figures 4.5a, 4.5b and 4.5c. This task is one of the tasks required by the IEEE 3DUI contest 2016. The goal is to pass a cube through a labyrinth while maximizing the courses filling of the object. First, we present the *Giant*'s interaction technique based on a bentray in Subsection 4.3.4.1. Then, we describe the *Ant*'s interaction technique in Subsection 4.3.4.2, and the handling of optional concurrent manipulation (for the object rotation and scale) in Subsection 4.3.4.3.

4.3.4.1 Global Manipulation

The user on the zSpace (*Giant*) can translate the object (actually the *Ant*'s conveyor in the IIVC model) with a bent ray inspired from the interaction technique proposed by Riege et al. [Riege et al., 2006]. The ray is controlled in position and rotation by the tracked stylus. One button is used for object grabbing, and the other buttons are used to switch between four point of views: front, left, back and right. The ray is bent during the object translation in order to respect three constraints:

3. <https://www.oculus.com/en-us/>

4. <http://www.razerzone.com/fr-fr/gaming-controllers/razer-hydra-portal-2-bundle>

5. <http://www.samsung.com/fr/galaxynote4/gear-vr/>

6. <https://www.google.com/get/cardboard/>

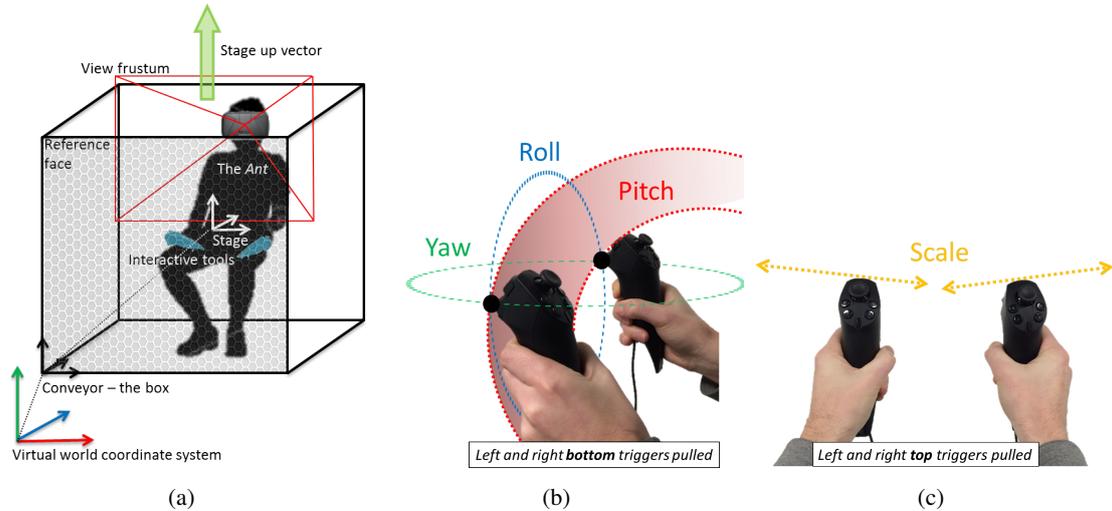


Figure 4.7: (a) The *Ant*'s IIVC instantiation. The front textured face of the object (here, a box) is the current stage front face that is used as reference for object manipulation performed by the *Ant*. The stage coordinate system is always centered in the object. (b) and (c) The two metaphors used by the *Ant* to rotate and scale the manipulated object.

- First, the manipulated object is attached to the ray extremity with a spring joint, as proposed by Fröhlich et al. [Fröhlich et al., 2000]. The physical collider of the object avoids it to pass through other objects, such as corridors. The ray is bent accordingly.
- Second, we manually limit the ray extremity speed when an object is grabbed. The goal is to not disturb the distant user inside the manipulated object and reduce his cybersickness.
- Third, a last constraint is optional. We added an active help for the translation. It is a *magnetic path* that represents the perfect path to follow. The manipulated object is also connected to the closest point on this path with a spring joint.

To make the others understand the *Giant*'s actions, his head, stylus and 3D ray are rendered in the shared environment as shown on the top Figure 4.8a.

4.3.4.2 Inside Object Manipulation

The *Ant* is placed inside the manipulated object. He can scale and rotate it with the two Razer Hydra controllers thanks to bimanual metaphors inspired from the work of Cutler et al. [Cutler et al., 1997]. These manipulations are performed with a fix reference: the object front face. This reference face can be changed. Figure 4.7a illustrates our instantiation of the IIVC model for the *Ant*, and especially explains the meaning of the reference face in the user stage according to the manipulated object. We propose a symmetric technique between both hands to switch the current stage front face based on joysticks. Up, bottom, right and left joystick triggers are used to apply a 90° rotation to the stage. An asymmetric aspect is introduced in order to turn around the normal current front face. For this purpose, we use the joystick button capability, to choose if we turn in the clockwise or the counter-clockwise direction, according to the hand used.

Manipulations of the conveyor are physically constrained, thus, the manipulated object can not pass through an obstacle. As shown in Figure 4.7b, the rotation is made with a modified version of the grab-and-twirl metaphor. Compared to the classical version, the pitch rotation is performed with a metaphor close to a plane yoke by orienting the two controllers to the top or

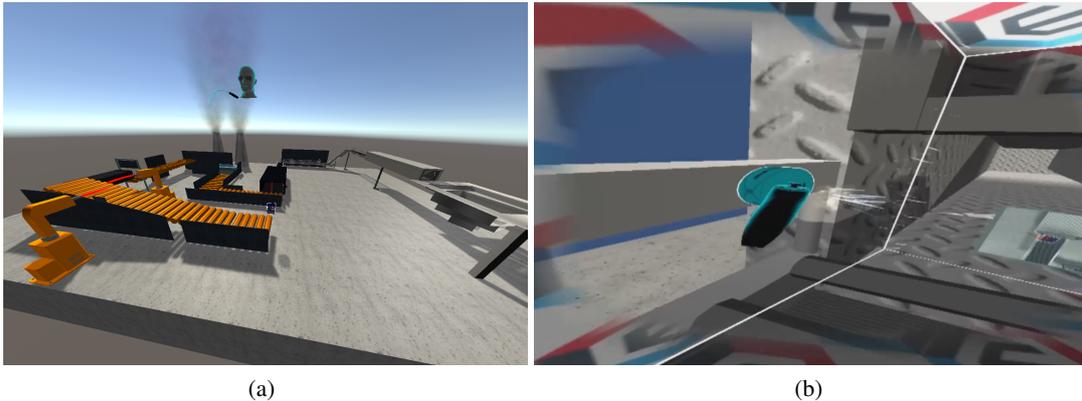


Figure 4.8: The virtual factory scene that provides scale-one interactions. Buttons are present in the scene to trigger different actions such as opening doors or painting the object. Triggering these actions is needed to fulfill the task. Therefore, as shown on (b), the *Ant* has direct manipulation capabilities, he can push a button by colliding it with one of his controllers.

to the bottom. The scale of the object is controlled with a grab-and-scale metaphor by bringing closer or further the two Razer Hydra controllers while pushing two corresponding buttons (cf. Fig. 4.7c). According to the selected mode, the object scale can be defined as uniform, based on the distance between the two controllers, or non-uniform by projecting the distances on the x,y and z axis of the *Ant*'s stage.

Two visual feedbacks are rendered to make the *Ant* understand the distance between the manipulated object and possible obstacles. First, we render particles at the collision points. Second, a virtual grid visible in blue at bottom in Figure 4.5b, parallel to the user current front face, is displayed outside of the object. This grid helps the user to perceive the distance between the manipulated object and other objects such as the courses. In addition to these feedbacks, we compute a filling ratio that is displayed in the integrated Graphical User Interface (GUI). It allows the *Ant* to be aware of the courses filling by the manipulated object, and to maximize it by using this quantitative ratio as well as the 3D visual feedbacks.

In some particular situations, local interactors can be integrated at scale one in the VE. For instance, in the example given in Figure 4.8b, buttons are available in the scene for triggering different actions such as opening doors. These buttons are too small for the *Giant*. Therefore, as the two Razer Hydra controllers are rendered into the VE, the *Ant* can use them to interact with these local interactors. Here, by touching one button with one of the controller, the *Ant* can trigger the corresponding action.

To guide the *Ant* when he is placed in a closed environment such as the provided labyrinth, different spatial cues can help him. They are shown in Figure 4.5b. First, a World-In-Miniature [Stoakley et al., 1995] shows a third person view that focuses on the manipulated object. Second, an arrow simulates a compass to show him the direction to follow. Third, in the labyrinth, the path to follow is also indicated with arrow signs.

For awareness issues, the viewpoint of the *Ant* is shown to the other users by displaying his frustum and stage up vector.

Reducing cybersickness due to motion As shown in Figures 4.5b and 4.8b, a progressive transparency effect is applied to the manipulated object from the screen extremities to the screen

center. This technique is adapted from the proposal described in [Lacoche et al., 2015b]. Here, it is used as an anti-cybersickness filter that aims to make the peripheral view of the user consistent with his head movement. Therefore, the user’s peripheral view is less disturbed by translations performed by the *Giant*. Some preliminary evaluations of this effect have been performed in another context and have shown good results.

This approach consists in applying a progressive alpha blending over the screen edges to the virtual objects for the two stereoscopic cameras. With this effect, when an object is close to a screen edge, its rendering is adjusted in the views of both the left and right eye. Indeed, we suppose that modifying only one view can lead to an increased visual discomfort and difficulty to fuse the two images.

We propose applying a linear decreasing alpha blending on the two axes of the screen. To apply the effect, we consider the screen as a space where all pixel coordinates take a value between (-1, -1) (bottom left corner) and (1, 1) (top right corner). The transparency is computed in order to get a “squircle” shape, a particular case of a superellipse. A square shape would have been coarse on the screen corners, and a circle shape would erase too much of the objects on these corners. For both cameras, for a pixel (x, y) , we compute its alpha factor in two steps. We first compute its distance to the squircle center. This computation is described in equation 4.1.

$$d(x, y) = \sqrt[4]{x^4 + y^4} \quad (4.1)$$

Second, a linear approach (Equation 4.2) uses this distance to compute the alpha factor. The distances that define minimal and maximal opacities are expressed in screen space. For our instantiation of this rendering technique, we have set the minimal distance to 0.8 and the maximal one to 0.97.

$$Alpha(x, y) = 1 - \frac{d(x, y) - distMinPlane}{distMaxPlane - distMinPlane} \quad (4.2)$$

4.3.4.3 Concurrent Manipulation

As optional collaboration possibilities, we provide concurrent manipulation capabilities for the rotation and the scale of the object.

Rotation We propose an optional rotation control scheme based on the concurrent action of the *Giant* and the *Ant*. In some particularly difficult circumstances, for instance if the *Ant* is completely lost and does not achieve to find the correct way to orient the object, the *Giant* can help him to rotate the object by acting on the *Ant*’s conveyor orientation. For this purpose, the *Giant* can use his bentray-based interaction technique as explained in scheme Fig. 4.9. The interaction starts when the bending of the ray is up to a specific threshold, i.e when \overrightarrow{BC} length is up to 2.0m in our implementation. Then, the manipulated object rotation is controlled based on velocity according to the \overrightarrow{BC} length, as follow:

```

Each frame
if  $\overrightarrow{BC}$ .length > threshold then
    angle =  $((\overrightarrow{BC}.length - threshold) * velocity * deltaTime)^2$ 
    object.rotateAround( $\overrightarrow{AB} \times \overrightarrow{BC}$ , angle)
end if

```

It results in a concurrent manipulation with the rotation capability of the *Ant* that is performed by integrating both actions with a growing *Giant*’s factor according to its ray bending.

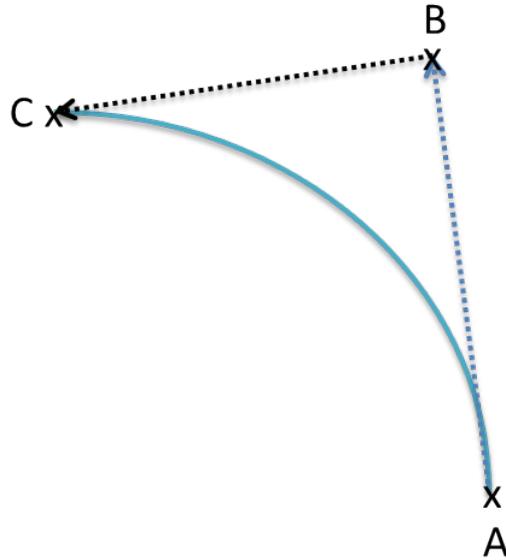


Figure 4.9: Scheme of the technique used to apply rotation to the object using the *Giant*'s bentray. A and B are respectively the beginning and the end of the straight, not rendered, ray. C is the end of the rendered bentray. Axis of rotation is computed as the cross product between \vec{AB} and \vec{BC} .

Scale The scale control is shared between the *Ant* and the user of the GearVR. To solve this concurrency, we add the factors that the two users want to apply to the scale. Thus, the fuse of the concurrent scale control is not based on an average method, but on a relative setting that allows synchronous manipulation without conflict. It means that each frame, each user asks to the scale manager component to increase or decrease the object scale by a vector. Then, this manager adds all the requests and sets the object scale accordingly if possible (i.e taking care of environmental physical constraints).

4.3.5 Implementation

Regarding the implementation of the prototype, Figure 4.10 illustrates the architecture of the application. A software overlay of SmartFox Server⁷ is used to manage collaboration. It ensures the synchronization of the shared VE between the different users as well as a consistent physical simulation. For rendering, scripting and managing the scene we use the Unity3D game engine⁸. The interaction part is independent from the devices and from the game engine used. Indeed, it is developed with an implementation of the plasticity models presented by Lacoche et al. [Lacoche et al., 2015a]. With this solution, each user automatically benefits from the adapted interaction techniques according to his available devices. These plasticity mechanisms also give us the possibility to easily exchange the current devices used. For instance, we could use an HTC Vive⁹ instead of the combination Oculus Rift / Razer Hydra for the *Ant* or any desktop environment instead of the Google Cardboard for the spectator.

7. <http://www.smartfoxserver.com/>

8. <https://unity3d.com/>

9. <https://www.htcvive.com/us/>

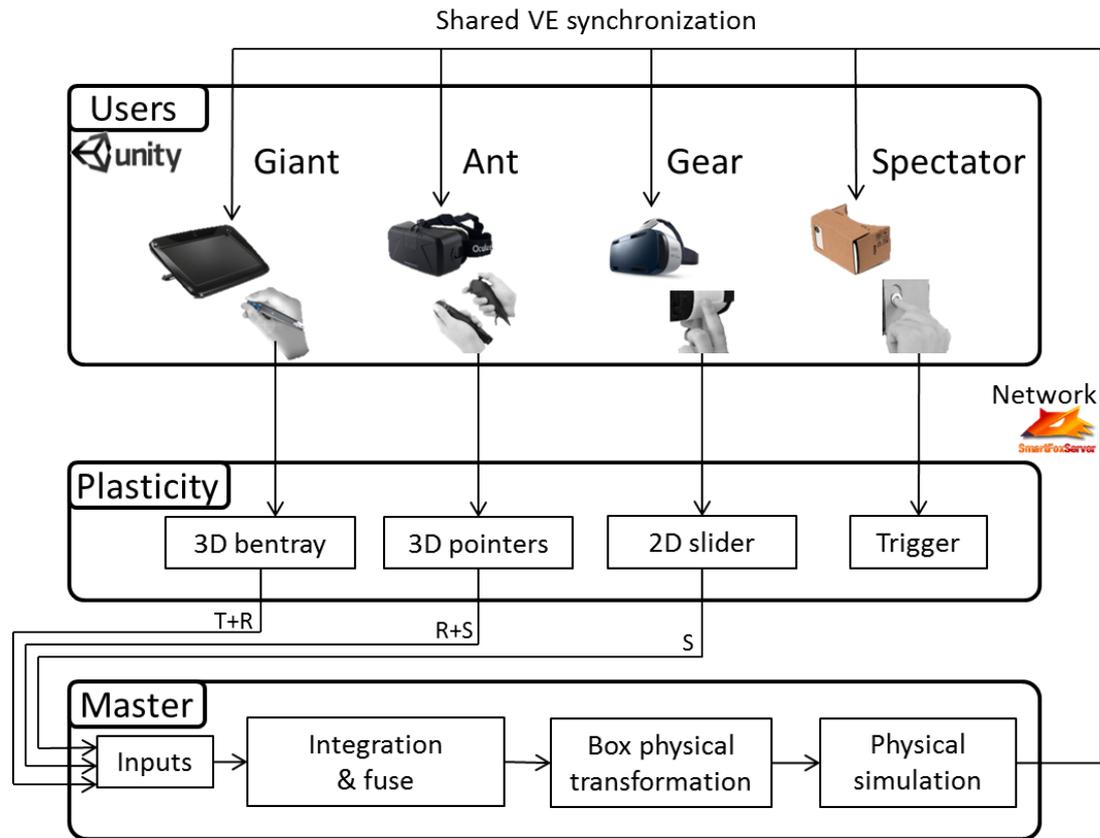


Figure 4.10: The global architecture of our implementation. T,R and S respectively refer to translation, rotation and scale.

4.3.6 Preliminary Tests

We did not perform any formal evaluation. However, we tested our approach with different users and scenarios. These first test users have experience with VR applications and 3D interactions. Regarding the commands of the *Ant*, we first tried the classical version of the grab-and-twirl metaphor in order to modify the manipulated object rotation. The different users did not feel comfortable with this interaction and they seemed to perform better with our interaction technique that reminds a plane yoke. Indeed, as the user is placed inside the object, controlling it as the user would control a vehicle seemed more natural. To continue, the different visual feedbacks for the *Ant* for making him understand his spatial relationship with the environment were added after multiple tests. Indeed, as the user did not know well the different labyrinths, they seemed to be relatively lost after multiple movements. For the *Giant*, we have compared the bent ray with a classical straight one without any speed limitation. This approach was really easy to understand for the *Giant* but it was really disturbing for the *Ant* to move as fast. Moreover, when the *Ant* was translated too fast, he did not have the time to anticipate the next obstacles and could not adapt his rotation and scale accordingly.

4.3.7 Conclusion & Perspectives

We proposed an asymmetric approach for co-manipulation in shared multi-scale environment in a specific context. In particular, it is useful to manipulate an object inside narrow corridors while maximizing its scale and providing direct interaction capabilities to one user, the *Ant*, with items of the VE. It is based on the collaboration between a *Giant* with a global view of the scene and an *Ant* immersed inside the manipulated object. Additional users can also be included to help the two main users in their task completion. Moreover, our approach benefits from a plasticity mechanisms that handle the automatic adaptation of the interaction technique according to the devices used by the different users with heterogeneous setups. This work has been submitted for the 3DUI Contest 2016, and we received the **honorable mention award** for this contribution.

Moreover, preliminary users tests show a good efficiency of the different interactions techniques, but a formal evaluation should be done in order to confirm the performances of the approach. First, it would be interesting to compare our proposal with a solution from the state-of-the-art where the collaboration is symmetrical, i.e with equivalent roles and viewpoints. Second, our solution includes an oral communication between the different participants. It would be interesting to evaluate the set of interaction techniques without this oral communication. As a result, we could include other awareness mechanisms in order to improve the collaboration when the participants can not talk with each other.

Conclusion

In this thesis, we presented research that focuses on remote collaboration using Mixed Reality (MR) technologies taking advantages of the natural human abilities in performing 3D tasks. In particular, this context introduces potential asymmetrical dimensions in the collaboration process.

The first one is the different roles that each collaborator can endorse. Indeed, in many cases collaborators can have a common task to achieve; however, each collaborator can lead subtasks according to the needed skills. This is a key feature of remote collaboration that allows users from different domains to work together in order to benefit from their own capabilities while not being physically co-located. Roles can also be intrinsically different if considering hierarchy in the collaboration, for instance with a trainer/teacher and a trainee/pupil.

The second asymmetrical dimension is introduced by the different Point of Views (PoVs) of collaborators. This asymmetry can be even more challenging if users belong to a multi-scale Virtual Environment (VE). Indeed, in order to provide an efficient collaboration, users must still be aware of others' perceptual and interactive capabilities. However, when sharing a multi-scale VE, users' capabilities often depend on their PoV since it impacts their abilities to perform some tasks that require a specific PoV. Thus, our work is also interested by this aspect that can greatly impact on the efficiency of such MR-based collaborative systems.

The third one is directly caused by the wide variety of MR devices. It leads to consider many different I/O devices used in a MR collaboration. Thus, in addition to the previous dimension, this one generates a heterogeneous collaboration with different interaction and perceptual capabilities induced from hardware limitations. Since the only way to interact and to get feedback from a simulation is to use I/O devices, their capabilities have a great impact on the levels of presence and awareness that a user is able to maintain. It requires to carefully design collaborative MR applications according to the different setups used.

The last asymmetrical dimension is directly linked to our context: collaboration in MR. MR is defined as a continuum from the real world (excluded) to a pure virtual world; between these two extrema, an application can provide several levels of virtuality depending on the amount of real and/or virtual content perceived by the user. For instance, remote collaborators can share a VE while a user is immersed in it using a Virtual Reality (VR) setup, while another user is equipped with an Augmented Reality (AR) setup to perceive the shared VE and to interact with its content. This variety of possibility in MR allows to imagine new kinds of paradigm to improve remote collaboration, but it also introduces complex issues to deal with in order to maintain a correct level of awareness between collaborators and their shared activities.

Contributions

We considered the previous asymmetrical dimensions introduced by remote collaboration in MR. Particularly, our work aims to design a way to describe such applications in terms of features responsible for maintaining awareness of his own and others activities. Indeed, these feedbacks are the signals that can be perceived by users and interpreted in order to maintain their own awareness of the shared VE. Thus, we introduced a model able to differentiate user activity in terms of intention into the collaboration (i.e if a user only modifies his own body state, or if a user directly interacts with shared content of the VE) and to describe feedbacks from the shared VE according to their sources and meanings (i.e if they are issued from VE affordance, interactions on virtual content or others collaborators activities). Then, we extended this model to enable the description of more complex situations taking into account more asymmetrical dimensions that are PoV (i.e in a multi-scale shared VE) and level of virtuality (i.e if collaborators share a VE with heterogeneous MR applications). To do so, we introduced the concept of personal virtual spaces that considers partial synchronization of a shared VE between remote collaborators; it manages that virtual content can be different if collaborators belong to a multi-scale VE and run different kinds of MR applications (for instance, a remote collaboration with an AR user and a VR user).

Moreover, we ran an exploratory user study in which we evaluated the impact of user representation on the awareness of others' activity in a multi-scale VE. It led to conclude that every user does not interpret the other user activity similarly and does not use available feedbacks in the same way. Particularly, we identified a group of subjects, called experts, that is able to adapt their strategies to estimate the other user activity according to the other user representation and available feedbacks. On the contrary, the other group, called novices, mainly selects a strategy that seems to fit according to their own interpretation of the situation, and does not change of strategy, that could fit better, when the situation is modified. It proves that, even if awareness is a very personal interpretation of a situation depending on his/her own experience and perception, users can reveal similar behaviors, in terms of strategy to build and maintain awareness, according to their own knowledge and experience of similar environments. Moreover, since experts are able to fit their strategy according to available feedbacks, they better handle the interpretation of different signals; and the design of interfaces should take this into account in order to provide enough feedbacks to improve awareness of experts, while not overloading novices, that could decrease their level of awareness due to an increased cognitive load, by using simpler interfaces.

Then, we address an actual application of remote collaboration in MR that is remote guiding. We were able to apply our previously defined awareness model to this use case in order to describe our two approaches. Our global approach is based on a remote expert equipped with a VR setup who assists a distant operator in performing a physical task through an AR interface. First, we experimented a setting inspired from our previous exploratory study with a multi-scale VE. The expert has a global view of the operator's workspace through a 3D reconstruction streamed from a camera moved by the operator. The expert is able to indicate items to pick and to point specific spots to act on it; the operator perceives this guidance with the AR display that manages a correct mix between virtual and real content in terms of co-location, relative depths and occlusions. Moreover, we developed a specific navigation guiding technique that takes advantage of our setting and reveals that visual continuity between the operator's display and the spot pointed by the expert improves reaching performances. Second, we go further by virtually collocating the remote expert into the operator's workspace at 1:1 scale. With this new setting, the expert can stand face-to-face to the operator, freely move into the shared VE and

even share the exact operator's PoV. Thus, we introduced the *Vishnu* paradigm that proposes to the expert to share the operator's PoV in order to guide him using virtual arms gestures. The main idea of this paradigm is to let the expert interacts into the shared VE through virtual arms; sharing the operator's PoV, the virtual arms appear to be co-located with both users; thus, from the operator's PoV, the *Vishnu*'s arms seem to come out his/her own shoulders in addition to his/her real arms. Using an Inverse Kinematic (IK) approach, the *Vishnu* paradigm provides a remote guiding system based on virtual arms gestures directly co-located with the operator's real body. It leads to drastically decrease the cognitive load because the visual guiding cues are directly mapped into the operator's field of view (FoV), and to improve expert's interaction capabilities by providing true manipulation capacities based on natural gestures. Moreover, we also proposed a navigation guiding technique based on the *Vishnu* paradigm. Its advantage lies in using existing virtual arms to guide the operator in reaching a specific spot while reducing his/her cognitive load. Inspired from the previous setting, this technique is based on a visual continuity between the operator's body and the pointed spot, by stretching the virtual arms from the operator's shoulders to the spot to walk. It benefits from less disturbance into the operator's FoV than other techniques because arms are displayed in the operator's peripheral vision, and not in front of him/her.

Last, we focused on a sensitive case in terms of users' behaviors synchronization that is virtual co-manipulation task. It requires that collaborators maintain a high level of awareness about others' activity to efficiently achieve the task. In this context, we proposed a generic approach that is able to handle many users with heterogeneous devices and asymmetric PoVs in a co-manipulation task. Then, we add the asymmetrical roles dimension in the process by providing a new hybrid approach based on a partial split of Degrees of Freedom (DoF) between collaborators adapted to their PoVs and their devices, while adding feedbacks to let them be aware of others' activity. Moreover, we demonstrated how our awareness model can describe this kind of scenarios with more complex asymmetrical roles in a pure VR remote collaboration. Our contribution on the hybrid approach to perform co-manipulation has received the *honorable mention award* at the IEEE 3DUI Contest 2016.

Perspectives

Our research leads to several contributions around the asymmetrical dimensions identified in MR-based remote collaboration. Still, we left several tracks of improvements about our proposals, and we developed some kind of forward vision about a possible future for this technological field.

First, about our proposal of an additional asymmetrical loop, developed in section 2.5.6, used to convey additional feedbacks about the user's analysis of some informational signals sent by another user; it would be interesting to go toward an automatic process able to manage the triggering of such feedbacks. It could lead to a system able to provide explicit acknowledgment to a user about the analysis of another user without any explicit user interaction. Particularly, in the context of our exploratory user study, the 2D mental rotation task has been deeply studied in the psychology domain, and we can estimate the time needed to complete such analysis based on the rotation of the shape to recognize. This analysis duration is pretty much the same for every human, and does not depend on some specific capabilities (unlike 3D rotation). Thus, we could consider a technique that would detect which objects are visible by a user (in the foveal vision, using eye-tracking technology for instance) and accordingly sent an acknowledgment to the other user when he/she spent enough time to analyze the 2D mental rotation to be able to solve

it. This particular task could be used to evaluate the feasibility of such a system. The next step would be to design a new approach able to integrate more advanced sensors and algorithms for behavioral analysis such as Brain-Computer Interface (BCI) and real-time analysis of body and eye tracking data. By estimating the activity of the user based on such analysis, the system could automatically and accurately add valuable feedbacks to collaborators about others activity. It could enable to generalize our proposal of asymmetrical awareness loop in more realistic context of remote collaboration.

Second, concerning our first setting managing remote guiding in MR (cf. section 3.4), we proposed to the expert to interact with a global PoV on the remote workspace. We could also imagine another situation in which the expert would be tiny in the shared VE in order to perceive more details needed to perform the diagnostic phase or even to propose more appropriate guiding indications to the operator in some specific situations. It could lead to a more efficient system with more interaction possibilities for the expert who could freely navigate into a personal (but still shared) multi-scale VE. It would require to design new kinds of navigation guiding technique in order to provide the expert with the ability to explain to the operator where to move the streamed camera to perceive tiny details of the remote workspace.

Last and not least, our *Vishnu* paradigm (cf. section 3.5) could be customized in several ways. First, for now, the design of the virtual upper and forearms are built with linked cylinders and spheres. Different rendering conditions could be investigate in order to evaluate the impact of the current design compared with robotic or realistic skinned arms for instance; in the same way the uncanny valley effect describes how level of realism of humanoids has an impact on their acceptability, thus on the whole performance of the human/robot or human/virtual agent collaborative process [Mori et al., 2012]. Therefore, the design of the *Vishnu*'s virtual arms could have a significant impact on the efficiency of the guidance as well as the acceptance of such a system that adds two virtual arms to the operator in addition to his/her real arms. Second, retargetting issues should be investigated in this specific context. Indeed, the *Vishnu* paradigm proposes to co-locate the virtual arms with both users' shoulders; however, their body dimensions are different, thus a specific IK algorithm handling retargetting of the expert's virtual arms guidance for the agent should be developed to manage properly this particular situation. Third, our forward vision about tele-maintenance naturally tends to explore this kind of systems that virtually immersed the remote expert into the operator's workspace at 1:1 scale. However, it should not restrain navigation capabilities of the expert, especially letting him freely navigate into a multi-scale shared VE seems to be a needed feature to cover most of possibly faced situations. Nevertheless, guidance at 1:1 scale with share PoV seems to be an efficient way to perform such tasks. Future systems could introduce the concept of body sharing based on waist co-location between the remote expert and the operator. Then, the expert could show which gestures to perform for arms and legs with natural perceptual capabilities for the operator. It could be used in many domains, such as for sport training [Han et al., 2016], or even body or gender exchange experiences [Bertrand et al., 2015]. Still, in order to be able to provide a high quality and fidelity user experience while erasing the physical distance between users, hardware technologies need major improvements. To our opinion, the main one is head-mounted AR displays. Indeed, nowadays, provided FoV is not large enough to cover the entire human FoV; and this piece of hardware is facing many other challenges to overcome: natural vision features such as depth of field and dynamic focus, or even rendering quality considering opacity of virtual content using see-through glasses. The last hardware challenges would certainly be to develop perfect inside-out tracking technology with no latency and to improve data transfer over the network, especially concerning large content, such as data from a very high-resolution RGB-D

camera (color and depth), that requires fast compression and streaming algorithms.

The day technical limitations will not be issues anymore, remote collaboration in MR could reinvent some of our ways of life and how we communicate, work, socialize and enjoy life experiences. In addition to be useful in industry, it will also offer new mediums to a general audience when its adoption will have proved the valuable features it can provide. To this end, user experience must be designed in a way it feels comfortable and natural to use. In this context, the feelings of presence and of co-presence are some key features to manage properly in order to achieve this goal. Thus, studies about awareness in this kind of environments must still be conducted to extend our knowledge about responsible mechanisms and strategies in order to work toward transparent remote collaboration even more efficient than real world local collaboration.

List of Figures

1.1	Scheme of the main asymmetrical dimensions found in remote collaboration in MR.	6
1.2	Illustrations of remote collaboration with asymmetric roles.	6
1.3	Illustrations of a sandbox providing collaborative remote interactions in a shared world.	7
1.4	MR devices examples.	8
1.5	Milgram’s continuum.	9
1.6	VR system examples.	10
1.7	AR and AV illustrations.	11
1.8	Illustrations of mid-air bimanual interaction schemes in semi-immersive systems.	12
1.9	Collaborative MR systems.	14
1.10	A summary of our contributions in three fields we focus on.	20
2.1	Illustrations of visual cues used to enhance awareness in CVE.	23
2.2	Heterogeneous collaboration and IIVC model.	25
2.3	The classical input/simulation/output loop describing an interactive system.	26
2.4	Interactive system model detailing types of user activity and feedback.	27
2.5	Interactive system model integrating user cognitive loop.	28
2.6	Our generic awareness model for CVE.	29
2.7	Integration of personal virtual spaces in our awareness model for multi-scale CVE.	31
2.8	Our model enables to describe asymmetrical MR collaboration.	32
2.9	Illustration of an asymmetric collaboration with two complementary PoVs.	33
2.10	Illustration of different PoVs available in a multi-scale VE.	34
2.11	The asymmetrical dimensions introduced in our preliminary evaluation about awareness in CVE.	36
2.12	(left) The visitor’s immersive view; (right) The guide’s global view.	37
2.13	Illustration of the experimental use case implementing our CVE model.	37
2.14	Illustration of our implementation of the safe-explicit feedback \square for this experiment from the guide’s point of view.	39
2.15	Difference of time between the visitor validation and the subject estimation: (a) globally, (b) when subject validated after the visitor, (c) and regarding the availability of \square .	41
2.16	Strategies applied according to the task and the expertise.	42
2.17	Successful applied strategies according to the task and the expertise.	43
2.18	Awareness loops in CVE.	47
3.1	Navigation guiding techniques.	51
3.2	Manipulation guiding techniques.	52

List of Figures

3.3	Description of our two approaches in terms of asymmetrical dimensions introduced in our systems.	53
3.4	Description of our HWD and WIM system instantiating our awareness model and merging virtual and real spaces.	55
3.5	Description of our HWD and WIM system instantiating our awareness model. . . .	56
3.6	Screenshots of the HWD and WIM prototype.	57
3.7	Illustration of the asymmetrical (HWD and WIM) prototype. The agent on the left is guided by the remote expert on the right.	58
3.8	The back of our AR tablet prototype.	59
3.9	Squared guiding arrow principle controlled by the expert's 3D ray in dynamic mode.	60
3.10	Illustration of the static mode for navigation guiding.	61
3.11	The two guiding techniques compared in the study.	62
3.12	The targets location into the floor.	63
3.13	(a,b,c) Boxplots of collected quantitative results. (d) Summary of qualitative appreciations.	64
3.14	A conceptual top view of the <i>Vishnu</i> paradigm.	66
3.15	Description of our fully collocated system instantiating our awareness model and merging virtual and real spaces.	67
3.16	Description of our fully collocated system instantiating our awareness model. . . .	68
3.17	Illustration of the system and the viewpoints of the agent (left) and the expert (right) in a motherboard assembly scenario.	69
3.18	An illustration of the remote expert's viewpoint standing in front of the distant agent 3D reconstruction.	71
3.19	Implemented setups: the expert on the left and the agent on the right.	71
3.20	Illustration of a subject performing the pilot user study and the virtual targets' spaces.	72
3.21	Illustration of a subject's PoV in mode 1.	73
3.22	Illustration of a subject's PoV in mode 2.	73
3.23	Results for the first layout L1.	74
3.24	Head's yaw angle for all iterations of all subjects.	75
3.25	Results for the second layout L2.	76
3.26	The stretchable guiding arms, controlled by the remote expert's location, from the agent's viewpoint.	77
3.27	Principle of the stretchable guiding arms technique.	78
3.28	Example of 10 targets defined for one condition.	79
3.29	Boxplots of the delay between the target hitting by the guide and the subject, their mean distance until the hit, and their final position and orientation differences. . . .	80
3.30	Synthesis of subjects' qualitative appreciations.	81
3.31	Principle of the guiding extension to improve final precision.	82
4.1	Asymmetrical dimensions available with our generic approach for co-manipulation. The gray hatchings mean that users are not required to use pre-established configurations but can choose amongst these all combinations.	87
4.2	Description of our generic approach for co-manipulation instantiating our awareness model.	88
4.3	Co-manipulation of a virtual object managed by our generic approach.	90
4.4	Asymmetrical dimensions introduced in our hybrid approach for co-manipulation. .	92
4.5	Collaborative manipulation of a virtual object based on an asymmetric setting. . . .	93

4.6	Description of our hybrid approach for asymmetric co-manipulation instantiating our awareness model.	95
4.7	The Ant's IIVC instantiation and interaction techniques.	97
4.8	The virtual factory scene that provides scale-one interactions.	98
4.9	Scheme of the technique used to apply rotation to the object using the <i>Giant's</i> bentray.	100
4.10	The global architecture of our implementation.	101
A.1	CollabMgr architecture overview.	118
A.2	Screenshot of the GUI login implemented in CollabMgr.	120
A.3	Screenshot of the GUI provided to debug applications based on CollabMgr.	121
A.4	Screenshot of the popped GUI defining shared object parameters.	122
A.5	Screenshot of the UnitySharedObject component.	123
A.6	Screenshot of the UnitySelectableSharedObject component.	124

List of Tables

2.1	The most applied and most successful strategies in all the conditions for experts and novices.	44
2.2	ROI of the strategies applied illustrating their global success.	45

List of Acronyms

DoF	Degree of Freedom
VE	Virtual Environment
MR	Mixed Reality
VR	Virtual Reality
AR	Augmented Reality
AV	Augmented Virtuality
FoV	Field of View
PoV	Point of View
CVE	Collaborative Virtual Environment
QoS	Quality of Service
I/O	Input/Output
GUI	Graphical User Interface
HUD	Head-Up Display
TRS	Translation-Rotation-Scale
P2P	Peer-to-peer
LAN	Local Area Network
WLAN	Wireless Local Area Network
WAN	Wide Area Network
FPV	First-Person Viewpoint
ROI	Return On Investment
HMD	Head-Mounted Display
HWD	Hand-Worn Display
IMU	Inertial Measurement Unit
RGB-D	Red/Green/Blue-Depth

Appendix A

The CollabManager Framework

The CollabManager Framework (aka CollabMgr) is a middleware used to connect a network communication engine with a game engine in a generic way in order to manage needed features to develop remote collaboration MR applications.

The current version of CollabMgr is based on SmartFox Server (SFS). But, the conception of the framework would allow another implementation that would use another server solution, such as RakNet.

CollabMgr for Unity is a wrapper of CollabMgr. It allows the synchronization of remote shared scene between several distant users. Its main features are:

1. Transform synchronization in possibly multi-scale VE
2. Physical behaviors run on the master and synchronize on all clients
3. Selectable objects locally managed and synchronize on all clients
4. Co-manipulation (average/split dof)
5. Custom application commands

Feature (1) is the basic requirement of remote collaborative applications with multi-scale ability needed to maintain consistency of the objects' transformations. Feature (2) handles the consistency of the physical simulation of the shared application. If runs on each clients, the physical simulation would diverge over time, thus our framework uses a centralized approach to compute physical behaviors of the whole shared VE. Feature (3) allows more reactivity with local interactions and feature (4) provides a built-in management of co-manipulation computation. Last, feature (5) provides a way to handle application-specific behaviors that are not built-in in our framework. By passing custom messages through the network, developers can easily add collaborative features to their MR applications using a provided simple registration and events generation process.

In addition to these basic features, CollabMgr provide additional built-in advanced features useful for asymmetric remote collaboration in MR:

1. Color and texture synchronization
2. Enable/disable simulation object (GameObject in Unity3D)
3. Asymmetric MR (AR/VR) enabled

Feature (1) and (2) are advanced synchronization possibilities not natively available in most of the existing frameworks. For instance, feature (1) can be used to stream video data over the network, such as RGB and depth maps to provide remote 3D reconstruction capability. Last,

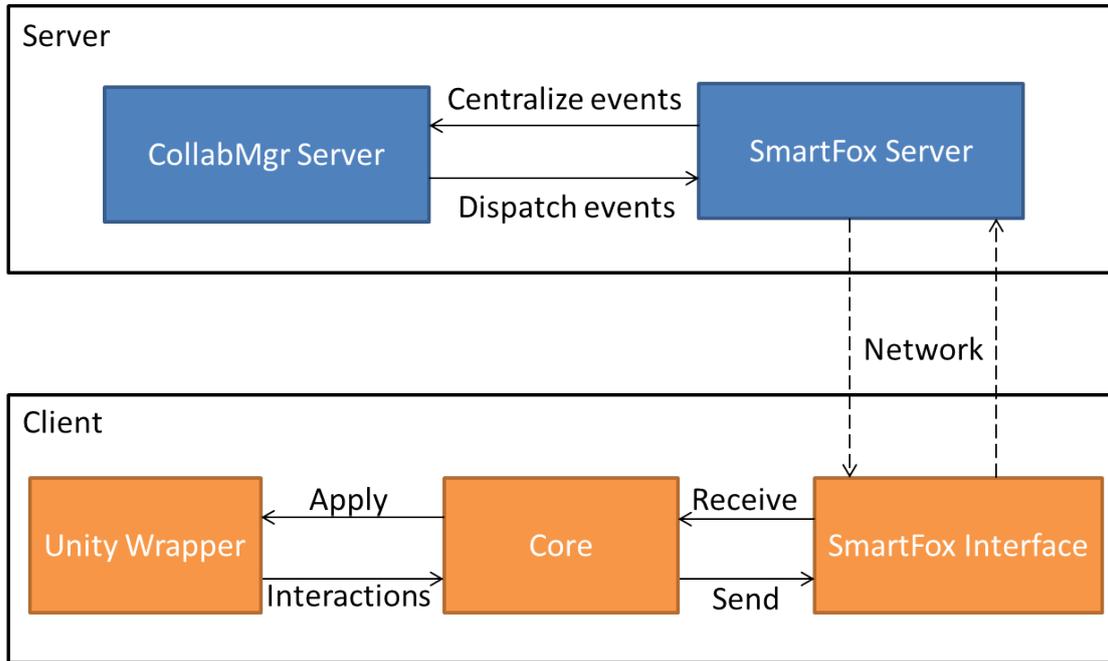


Figure A.1: CollabMgr architecture overview.

feature (3) is very specific to applications mixing AR for a user, and VR for another one. In this case, we can imagine cloned objects from the real world replicated in the VE as virtual objects. This leads to consider two different status for a same object: the real one manipulated by the AR user and the virtual one manipulated by the VR user. Each user must be able to see the object he can manipulate as well as the object the other can manipulate. We natively manage this asymmetrical aspect in our framework in order to ease the development of such applications.

A.1 Architecture overview

CollabMgr for Unity is composed of 3 main modules and 2 additional modules running on the centralized server, as illustrated on Figure A.1.

First, let describe the server part running on a single PC. It is composed of the SmartFox Server (SFS). It manages the creation of zones and rooms to provide the ability to have multiple simulations centralized on the same PC. It also provides a web interface to set advanced network settings and is able to recover from network failures in a transparent way in some circumstances. Each room must load a SFS extension (written in Java or Python), here called *CollabMgr Server*. In our case, this extension is in charge to dispatch every network events sent to the SFS Server. According to the types of the events, they can be dispatch either to one, several or all the clients (e.g using unicast, multicast or broadcast policy).

Second, each users of our framework is seen as a client. However, we distinguish two kinds of client. The first one, called the *master*, is the unique client which takes centralized decisions inside his *core* component. Generally, it is launched on the same PC as the server part, but it can also be launched on another one. Moreover, it can be a graphical client (e.g taking part in the simulation as other users), or it can be a non-graphical user used only for its centralization purpose. Other users are regular users.

The client's core component, written in C# is in charge of login process, user status handling, and virtual content synchronization according to the developer-chosen policy. The SmartFox Interface is the link between the independent core and the SmartFox Engine used for the network layer. An advantage of the SmartFox Engine is the amount of covered programming languages: its client API is available in ActionScript3, C#, Objective-C, HTML5, Java and C++. Thus it can be adapted to almost every existing platform: Flash and web browsers, .Net, Mono and Windows Universal, OSX, iOS and Android, JAVA SE/EE. Our implementation uses C# to ease the interfacing with our core component. Moreover, as for now we support Unity3D as game engine, we also used C# to develop the game engine-specific wrapper that integrates CollabMgr into Unity3D. The game engine is in charge of rendering and user input and output. Moreover, in our case, Unity3D also includes a built-in physics engine (based on Nvidia PhysX engine) that we used to simulate physical behaviors into the simulation. This physical simulation is only computed on the master node that is in charge to update other regular nodes that do not locally compute any physical behaviors. At the contrary, when manipulating a virtual object, it is the owner node of the object (e.g the node which first selects the object) which is in charge of updating every other nodes about the modifications of the manipulated object. This migration mechanism allows more reactive local interactions for the user who currently manipulates an object.

A.2 Documentation: How to use CollabMgr for Unity3D?

Setup a shared sync scene The minimum requirement in the graph scene is to put the CollabMgr prefab. It contains several scripts:

- UnityUser
- UnityGUILogin&ColorPicker / LoginConfigParser
- UnityCollabMgrGUI
- UnityGUIPickerInfos

and a child that contains UnityUsersPanel. We are going to describe the function of each component.

UnityUser defines the user parameters as well as the necessary network parameters. Actually, it contains all the settings needed to establish a connection with the server. The name has to be unique between each user because it is used as a unique ID. `Host`, `zone`, `room` and `port` are the network settings. For now, CollabMgr is implemented based on SmartFox Server (SFS). The host address must be the address of the machine that runs the SFS. `Status` is a choice between `Master` and `Client`. The first logged user must choose `Master` while the next ones will choose `Client`. `Color` is the user color. It can be anything. `Add User Representation`, if checked, will add the `Frustum` gameobject at startup to symbolize the user in the virtual environment. `Main Camera` references the main camera of the scene. `Display GUIuser`, if checked, will display a small info panel at the bottom of the screen with a background of the color defined with `Gui Label Background Color`.

In almost every case, all the connection parameters will be overridden at startup by the loading of the configuration (either through the GUI login, or the configuration file). Thus, it is not necessary to set them in the Unity Editor.

UnityGUILogin handles the login phase at startup. `Basic Skin` and `Background Color` are GUI settings without consequences. It references the `SceneMgr` whose purpose is discussed

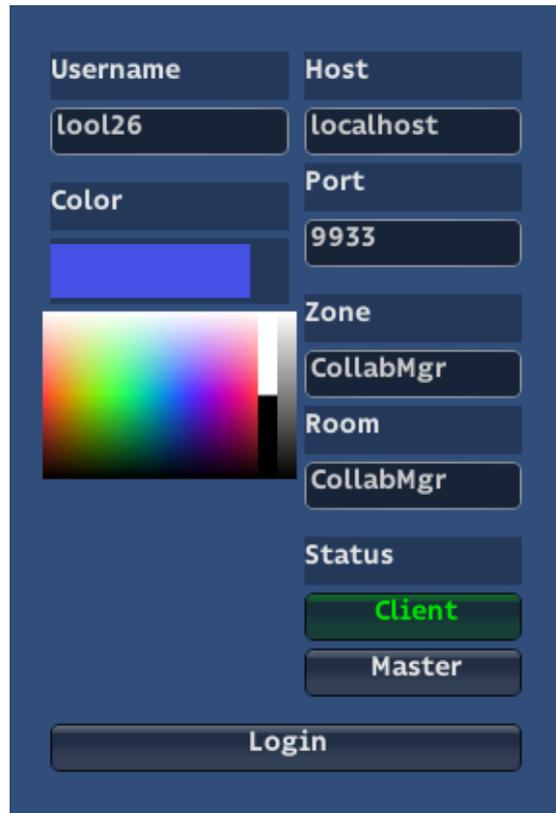


Figure A.2: Screenshot of the GUI login implemented in CollabMgr.

later. Others is a list of shared synchronized object that are not child of the SceneMgr. Every shared object that is not a child of the SceneMgr must be listed here. ColorPicker is a component that handles the color picker to choose the user color before login.

Notice that in case we do not want a GUI to login (as illustrated in Fig. A.2), we can use a configuration file thanks to the component LoginConfigParser. In this case, UnityGUILogin and ColorPicker must be removed. On Android, the configuration file must be deploy in the application directory on the sdcard.

UnityCollabMgrGUI is a useful tool to debug applications with GUI. It provides a hidden button (defined with the rectangle RButton and proportional to the screen size). When activated, it displays a 2 columns GUI, as illustrated in figure A.3. The left one lists the connected users, and the right ones the shared objects. A search bar is available on top of the column. You can click an object to open a window that lists its parameters. Moreover, a setting button is present and open a window to set some user parameters at runtime, including global co-manipulation policy per user (e.g average method or controlled DoFs).

UnityGUIPickerInfos is a complementary debug tool that allows to display the infos window of a shared object clicking on it (in addition to a key down). Moreover, it allows the setting of some manipulation parameters.



Figure A.3: Screenshot of the GUI provided to debug applications based on CollabMgr.

Users Panel is a GUI panel on the side of the screen that lists the connected users. The gameobject can be deactivated if it is not needed.

The SceneMgr Another useful prefab is the Scene that contains 2 components.

UnitySceneMgr is used at initialization to automatize the synchronization of its children without any other setting on the specific objects. Children tagged with either 'shared' or 'selectableShared' will become synchronized. It is also used to add new shared object at runtime. To do so, you must call `UnitySceneMgr.instance.addNewSharedObj(modelName, sharedType, parent, color, initpos, initrot, initscale, uselocal, isupdater, postFunc)`. The `postFunc` parameter is very useful since it allows to implement a custom process after the adding of the new shared object.

ApplicationCmdHandler handles the application command over the network. Actually, it can be placed on any gameobject, in case you do not need a Scene prefab. To add a new application command, you have to edit the script and add your new command as the examples. Then, the scripts that will have to receive the command must subscribe to the new event you created. Everything else is transparent. To send an application command, call `ApplicationCmdHandler.instance.sendCmd(user, sharedObj, customParam)`. Except `cmd` parameter, the others can be null, or you can send any data that you want, such as a user (yourself or another, the receiver for instance, or anything else you need), a shared object (the one you want to act on for instance), anything



Figure A.4: Screenshot of the popped GUI defining shared object parameters.

else using the `customParam`. The way you handle the command will define what you need to send with.

Shared objects synchronization It exists 2 kind of synchronized objects:

- `UnitySharedObject` used to synchronize physical objects that can not be selected
- `UnitySelectableSharedObject` used to synchronize any object that is selectable

Notice that children of the `SceneMgr` will be synchronized with default settings. If you want to customize parameters, you have to add yourself one of these 2 scripts to any object (children of the `SceneMgr` or not) and set it. Notice that in this case, no tag is needed.

UnitySharedObject Basically, this script handles the synchronization of the transform. Since the object is not selectable, it can be used in different situations:

- Physical object
- Input synchronization: frustum with HMD, sketcher with 6DOF,...
- Animated object

You can check what you want to synchronize, and in local or not. The basic policy let the Master user update everything. You can change this using `ForceIsUpdater` and `ForceNotUpdater` for specific scene settings. `AnimForceUpdate` can be used in scripts for particular cases. Figure A.5 illustrates the available Unity component.

Others functionalities are available through scripting:

- Color sync

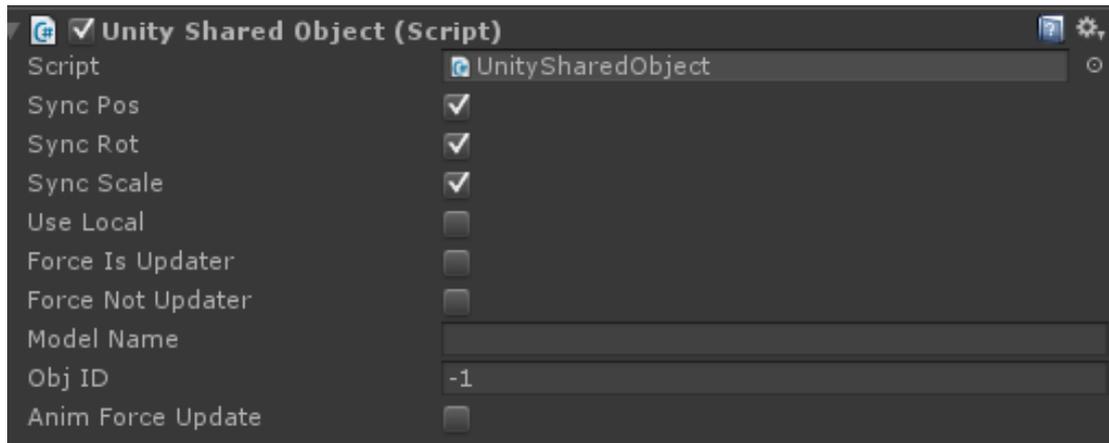


Figure A.5: Screenshot of the UnitySharedObject component.

- Texture sync
- Reparent sync
- Removing sync

They must be used through the SharedObject contained in the UnitySharedObject. For instance using `gameobject.GetComponent<UnitySharedObject>.getSharedObj().remove()`.

UnitySelectableSharedObject This script derived from `UnitySharedObject` and is used to synchronize selectable objects and manages the selection policy. In addition to shared objects, it adds 3 parameters:

- Selection Policy: `single` (only one user can select an object at the same time) or `multi` (enable co-manipulation)
- CoManip Policy: `average` (mean method) or `splitDof`. If `multi` enables, must not be set to `disable`.
- CoManip Dof: In case of `multi` and `splitDof`, defines the user dof controls. Notice that, for now, everything has not been tested, but the most relevant configuration is to let a user controls `Txyz` and another `Rxyz`. This configuration is tested and functional.

Figure A.6 illustrates the available Unity component.

These manipulation parameters can also be set at runtime through the settings (globally for all selectable objects) or the picker window (for a specific object).

Notice that the `Comanip Dof` parameter must be set specifically for each user since it defines the user dof controls, it is not a global shared parameter.

Example scenes The Unity package provides example scenes that demonstrate the use of most common features handled by the CollabMgr framework.

TestScene Children of the `SceneMgr` become synchronized and selectable at startup. Except the light that is animated and synchronized but not selectable. The `BearRotator` shows an example of command application. `NoisyTexture` is an example of texture synchronization. `ObjGen` is an example of adding new shared object at runtime. The scene also contains two customized shared object that can be co-manipulated.

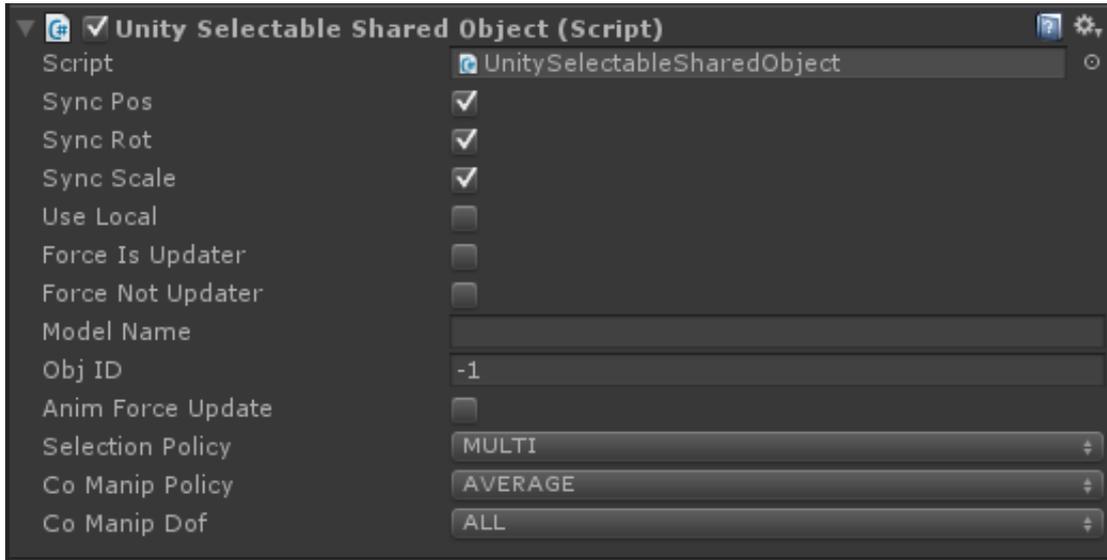


Figure A.6: Screenshot of the UnitySelectableSharedObject component.

WebcamStream This example synchronizes a texture from a camera stream. It can be used to stream the camera of an Android device to a computer and display the live stream on the plane. To be able to do that, notice that the gameobject contains a `UnitySharedObject` in addition of the `WebcamStream` component. Actually, it is a selectable plane, thus the synchronization component is a `UnitySelectableSharedObject`.

First installation First, install SFS 2X in a directory where you have all access rights, e.g. to write without to be an administrator. Go to the installation directory, then `SFS2X/extensions`. Create `CollabMgr` directory and copy `SFSCollabMgrExtension.jar` (found at the root of the project) in here. Access the web interface through your web browser at `localhost:8080`. Launch the Administration Tool 2X. Connect into it: Host: `localhost` ; Username: `sfsadmin` ; Password: `sfsadmin`. Go to the Server Configurator panel (hidden on the left side of the interface). Add the required socket addresses: the ip of the machine in both TPC and UDP. Go to the Zone Configurator panel. Create a new zone named `CollabMgr` (settings: Use custom log=false, Force logout=true, Overriden user maximum idle time=1000000) and add a new room named `CollabMgr`. Set the room extension: Name=`CollabMgr`, Type=`Java`, Main class=`sfsExtensions.SFSCollabManagerExtension`, and submit.

Regular use Just run SFS (`sfs2x-standalone.exe`) and check there is no error on startup. In addition, check the correct socket addresses are listened as you expect it, and the zone, the room and the extension have been loaded.

Possible errors:

- SFS can be locked the first launch and ask you to delete `SFS2X/Config/ExtensionManagerLock.txt`

A.2. DOCUMENTATION: HOW TO USE COLLABMGR FOR UNITY3D?

```
.....
>> Zone: CollabMgr
.....

11:32:40,497 INFO [main] managers.SFSRoomManager - Room created: { Zone: CollabMgr },
[ Room: CollabMgr, Id: 1, Group: default, isGame: false ]
11:32:40,506 INFO [main] Extensions - {CollabMgr}: ## SFSCollabManagerExtension
initialized
11:32:40,507 INFO [main] managers.SFSZoneManager -

11:32:42,427 INFO [main] v2.SmartFoxServer - Listening Sockets: { 127.0.0.1:9933,
(Tcp) } { 127.0.0.1:9933, (Udp) } { 10.10.255.132:9933, (Tcp) } { 10.10.255.132:9933, (Udp)
}
11:32:42,429 INFO [main] v2.SmartFoxServer -

  _____
 |  _  |  _  |  _  |  _  |  _  |
 |  _  |  _  |  _  |  _  |  _  |
 |  _  |  _  |  _  |  _  |  _  |
  _____

  _____
 |  _  |  _  |  _  |  _  |  _  |
 |  _  |  _  |  _  |  _  |  _  |
 |  _  |  _  |  _  |  _  |  _  |
  _____

[ 2.8.2 ]

11:32:42,435 INFO [main] v2.SmartFoxServer - SmartFoxServer 2X (2.8.2) READY!
```


Publications

International Conferences

- *Dealing with Frame Cancellation for Stereoscopic Displays in 3D User Interfaces*
Jérémy Lacoche, Morgan Le Chénéchal, Sébastien Chalmé, Jérôme Royan, Thierry Duval, Valérie Gouranton, Eric Maisel, Bruno Arnaldi, in Proceedings of IEEE Symposium on 3D User Interfaces (3DUI), Arles France, March 2015.
- *Toward an Enhanced Awareness in Asymmetric CVE*
Morgan Le Chénéchal, Sébastien Chalmé, Thierry Duval, Jérôme Royan, Valérie Gouranton, Bruno Arnaldi, in Proceedings of the International Conference on Collaboration Technologies and Systems (CTS), Atlanta Georgia USA, June 2015.
- *The Stretchable Arms for Collaborative Remote Guiding*
Morgan Le Chénéchal, Thierry Duval, Valérie Gouranton, Jérôme Royan, Bruno Arnaldi, in Proceedings of the 25th International Conference on Artificial Reality and Telexistence and the 20th Eurographics Symposium on Virtual Environments (ICAT-EGVE), Kyoto Japan, October 2015.

International Workshops

- *From 3D Bimanual Toward Collaborative Interaction Techniques: An Awareness Issue*
Morgan Le Chénéchal, Bruno Arnaldi, Thierry Duval, Valérie Gouranton, Jérôme Royan, in Proceedings of IEEE International Workshop on Collaborative Virtual Environments (3DCVE), Minneapolis Minnesota USA, March 2014.
- *Vishnu: Virtual Immersive Support for HelpiNg Users - An Interaction Paradigm for Collaborative Remote Guiding in Mixed Reality*
Morgan Le Chénéchal, Thierry Duval, Valérie Gouranton, Jérôme Royan, Bruno Arnaldi, in Proceedings of IEEE International Workshop on Collaborative Virtual Environments (3DCVE), Greenville South Carolina USA, March 2016.
- *When the Giant meets the Ant: An Asymmetric Approach for Collaborative and Concurrent Object Manipulation in a Multi-Scale Environment*
Morgan Le Chénéchal, Jérémy Lacoche, Jérôme Royan, Thierry Duval, Valérie Gouranton, Bruno Arnaldi, in Proceedings of IEEE International Workshop on Collaborative Virtual Environments (3DCVE), Greenville South Carolina USA, March 2016.

Poster and Demonstration Papers

- *Poster: Bimanual Design of Deformable Objects Thanks to the Multi-tool Visual Metaphor*
Morgan Le Chénéchal, Maud Marchal, Bruno Arnaldi, in Proceedings of IEEE Symposium on 3D User Interfaces (3DUI), Minneapolis Minnesota USA, March 2014.
- *Demo: Laying out Spaces with Virtual Reality*
Morgan Le Chénéchal, Jérémy Lacoche, Cyndie Martin, Jérôme Royan, in Proceedings of IEEE International Conference on Virtual Reality (VR), Arles France, March 2015.
- *Contest demo: When the Giant meets the Ant: An Asymmetric Approach for Collaborative Object Manipulation*
Morgan Le Chénéchal, Jérémy Lacoche, Jérôme Royan, Thierry Duval, Valérie Gouranton, Bruno Arnaldi, in Proceedings of IEEE Symposium on 3D User Interfaces (3DUI), Greenville South Carolina USA, March 2016.

Bibliography

- [Achibet et al., 2014] Achibet, M., Marchal, M., Argelaguet, F., and Lécuyer, A. (2014). The Virtual Mitten: A novel interaction paradigm for visuo-haptic manipulation of objects using grip force. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 59–66. IEEE.
- [Agrawala et al., 1997] Agrawala, M., Beers, A. C., McDowall, I., Fröhlich, B., Bolas, M., and Hanrahan, P. (1997). The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 327–332. ACM Press/Addison-Wesley Publishing Co.
- [Aguerreche et al., 2009a] Aguerreche, L., Duval, T., Arnaldi, B., et al. (2009a). A description of a dialog to enable interaction between interaction tools and 3d objects in collaborative virtual environments. *VRIC 2009*, pages 63–73.
- [Aguerreche et al., 2010] Aguerreche, L., Duval, T., and Lécuyer, A. (2010). Reconfigurable tangible devices for 3d virtual object manipulation by single or multiple users. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, pages 227–230.
- [Aguerreche et al., 2009b] Aguerreche, L., Duval, T., Lécuyer, A., et al. (2009b). Short paper: 3-hand manipulation of virtual objects. *JVRC 2009*.
- [Andujar et al., 2006] Andujar, A., Fairen, M., and Argelaguet, F. (2006). A cost-effective approach for developing application-control guis for virtual environments. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 45–52. IEEE.
- [Arnaldi et al., 2006] Arnaldi, B., Fuchs, P., and Guiton, P. (2006). Introduction à la réalité Virtuelle. In *Les Applications de la Réalité Virtuelle*, volume 4 of *Le Traité de la Réalité Virtuelle*. Les Presses de l’Ecole de Mines de Paris.
- [Azuma et al., 2001] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47.
- [Beck et al., 2013] Beck, S., Kunert, A., Kulik, A., and Froehlich, B. (2013). Immersive Group-to-Group Telepresence. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):616–625.
- [Benford et al., 1994] Benford, S., Bowers, J., Fahlén, L. E., and Greenhalgh, C. (1994). Managing Mutual Awareness in Collaborative Virtual Environments. In *Proceedings of the Conference on Virtual Reality Software and Technology, VRST ’94*, pages 223–236, River Edge, NJ, USA. World Scientific Publishing Co., Inc.

BIBLIOGRAPHY

- [Benford and Fahlén, 1993] Benford, S. and Fahlén, L. E. (1993). Awareness, Focus, and Aura: A Spatial Model of Interaction in Virtual Worlds. In *HCI (2)*, pages 693–698.
- [Bertrand et al., 2015] Bertrand, P., Gonzalez-Franco, D., Cherene, C., and Pointeau, A. (2015). *The Machine to be Another: Embodiment performance to promote empathy among individuals*.
- [Billinghurst et al., 2002a] Billinghurst, M., Cheok, A., Prince, S., and Kato, H. (2002a). Real world teleconferencing. *Computer Graphics and Applications, IEEE*, 22(6):11–13.
- [Billinghurst et al., 2002b] Billinghurst, M., Kato, H., Kiyokawa, K., Belcher, D., and Poupyrev, I. (2002b). Experiments with face-to-face collaborative AR interfaces. *Virtual Reality*, 6(3):107–121.
- [Billinghurst et al., 2001] Billinghurst, M., Kato, H., and Poupyrev, I. (2001). The MagicBook: a transitional AR interface. *Computers & Graphics*, 25(5):745–753.
- [Boas, 2013] Boas, Y. (2013). Overview of virtual reality technologies. In *Interactive Multimedia Conference. Southampton*.
- [Borst and Indugula, 2006] Borst, C. W. and Indugula, A. P. (2006). A Spring Model for Whole-Hand Virtual Grasping. *Presence: Teleoperators & Virtual Environments*, 15:47–61.
- [Bouville et al., 2015] Bouville, R., Gouranton, V., Boggini, T., Nouviale, F., and Arnaldi, B. (2015). # FIVE: High-Level Components for Developing Collaborative and Interactive Virtual Environments. In *Eighth Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS 2015), conjunction with IEEE Virtual Reality (VR)*.
- [Bowman and Hodges, 1997] Bowman, D. A. and Hodges, L. F. (1997). An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff. ACM.
- [Bowman et al., 2004] Bowman, D. A., Kruijff, E., LaViola, J. J., and Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- [Broll et al., 2004] Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., Mottram, C., Fatah gen Schieck, A., and Strothman, A. (2004). Arthur: A collaborative augmented environment for architectural design and urban planning.
- [Cai and Yu, 2014] Cai, G. and Yu, B. (2014). Event-based awareness promotion for distributed collaborative activities. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 302–309. IEEE.
- [Card et al., 1986] Card, S. K., Moran, T. P., and Newell, A. (1986). The model human processor: An engineering model of human performance. *Handbook of Human Perception*, 2.
- [Casera and Kropf, 2010] Casera, S. and Kropf, P. (2010). Collaboration in scientific visualization. *Advanced Engineering Informatics*, 24(2):188–195.
- [Chellali et al., 2010] Chellali, A., Dumas, C., and Milleville, I. (2010). Haptic communication to enhance collaboration in virtual environments. In *Proceedings of the 28th Annual European Conference on Cognitive Ergonomics*, pages 83–90. ACM.

- [Chevaillier et al., 2012] Chevaillier, P., Trinh, T.-H., Barange, M., De Loor, P., Devillers, F., Soler, J., and Querrec, R. (2012). Semantic modeling of Virtual Environments using MASCARET. In *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2012 5th Workshop on*, pages 1–8.
- [Claude et al., 2015] Claude, G., Gouranton, V., and Arnaldi, B. (2015). Roles in Collaborative Virtual Environments for Training. In *International Conference on Artificial Reality and Telexistence Eurographics Symposium on Virtual Environments (2015)*, pages 1–8.
- [Cooper, 1975] Cooper, L. A. (1975). Mental rotation of random two-dimensional shapes. *Cognitive psychology*, 7(1):20–43.
- [Cruz-Neira et al., 1993] Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A. (1993). Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM.
- [Cutler et al., 1997] Cutler, L. D., Fröhlich, B., and Hanrahan, P. (1997). Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114.
- [Dagnelie, 2011] Dagnelie, G. (2011). *Visual prosthetics: physiology, bioengineering, rehabilitation*. Springer Science & Business Media.
- [Dou and Fuchs, 2014] Dou, M. and Fuchs, H. (2014). Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In *Virtual Reality (VR), 2014 IEEE*, pages 39–44. IEEE.
- [Dourish and Bellotti, 1992] Dourish, P. and Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114. ACM.
- [Dupont et al., 2010] Dupont, F., Duval, T., Fleury, C., Forest, J., Gouranton, V., Lando, P., Laurent, T., Lavoué, G., and Schmutz, A. (2010). Collaborative scientific visualization: the collaviz framework. In *JVRC 2010 (2010 Joint Virtual Reality Conference of EuroVR-EGVE-VEC)*.
- [Duval and Fleury, 2009] Duval, T. and Fleury, C. (2009). An asymmetric 2d pointer/3d ray for 3d interaction within collaborative virtual environments. In *Proceedings of the 14th international Conference on 3D Web Technology*, pages 33–41. ACM.
- [Duval et al., 2008] Duval, T., Fleury, C., Nouailhas, B., and Aguerreche, L. (2008). Collaborative exploration of 3d scientific data. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 303–304. ACM.
- [Duval et al., 2006a] Duval, T., Lécuyer, A., and Thomas, S. (2006a). Skewer: a 3d interaction technique for 2-user collaborative manipulation of objects in virtual environments. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 69–72.
- [Duval et al., 2013] Duval, T., Nguyen, T. T. H., Fleury, C., Chauffaut, A., Dumont, G., and Gouranton, V. (2013). Improving Awareness for 3d Virtual Collaboration by Embedding the Features of Users’ Physical Environments and by Augmenting Interaction Tools with Cognitive Feedback Cues. *Journal on Multimodal User Interfaces*.

- [Duval et al., 2006b] Duval, T., Zammar, C., et al. (2006b). Managing Network Troubles while Interacting within Collaborative Virtual Environments. In *CSAC*, pages 85–94.
- [Fleury et al., 2010a] Fleury, C., Chauffaut, A., Duval, T., Gouranton, V., and Arnaldi, B. (2010a). A Generic Model for Embedding Users’ Physical Workspaces into Multi-Scale Collaborative Virtual Environments. In *ICAT 2010*, Adelaide, Australia.
- [Fleury et al., 2010b] Fleury, C., Duval, T., Gouranton, V., and Arnaldi, B. (2010b). A New Adaptive Data Distribution Model for Consistency Maintenance in Collaborative Virtual Environments. In *Proc. of JVRC*, pages 29–36.
- [Fraser et al., 1999] Fraser, M., Benford, S., Hindmarsh, J., and Heath, C. (1999). Supporting awareness and interaction through collaborative virtual interfaces. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 27–36.
- [Fraser et al., 2004] Fraser, M., Hindmarsh, J., Benford, S., and Heath, C. (2004). Getting the picture: Enhancing avatar representations in collaborative virtual environments. In *Inhabited Information Spaces*, pages 133–150. Springer.
- [Frécon and Stenius, 1998] Frécon, E. and Stenius, M. a. (1998). DIVE: A scaleable network architecture for distributed virtual environments. *Distributed Systems Engineering*, 5(3):91.
- [Freeman et al., 2003] Freeman, D., Slater, M., Bebbington, P. E., Garety, P. A., Kuipers, E., Fowler, D., Met, A., Read, C. M., Jordan, J., and Vinayagamoorthy, V. (2003). Can virtual reality be used to investigate persecutory ideation? *The Journal of nervous and mental disease*, 191(8):509–514.
- [Frey et al., 2008] Frey, D., Royan, J., Piegay, R., Kermarrec, A.-M., Anceaume, E., and Le Fessant, F. (2008). Solipsis: A decentralized architecture for virtual environments. In *1st International Workshop on Massively Multiuser Virtual Environments*.
- [Fröhlich et al., 2000] Fröhlich, B., Tramberend, H., Beers, A., Agrawala, M., and Baraff, D. (2000). Physically-based manipulation on the responsive workbench. In *Virtual Reality, 2000. Proceedings. IEEE*, pages 5–11. IEEE.
- [Fuchs et al., 2011] Fuchs, P., Moreau, G., and Guitton, P. (2011). *Virtual reality: concepts and technologies*. CRC Press, Inc.
- [García et al., 2008] García, A. S., Molina, J. P., Martínez, D., and González, P. (2008). Enhancing collaborative manipulation through the use of feedback and awareness in CVEs. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, page 32.
- [Gaucher et al., 2013] Gaucher, P., Argelaguet, F., Royan, J., and Lécuyer, A. (2013). A novel 3d carousel based on pseudo-haptic feedback and gestural interaction for virtual showcasing. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 55–58.
- [Gauglitz et al., 2014a] Gauglitz, S., Nuernberger, B., Turk, M., and Höllerer, T. (2014a). In touch with the remote world: remote collaboration with augmented reality drawings and virtual navigation. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 197–205. ACM.
- [Gauglitz et al., 2014b] Gauglitz, S., Nuernberger, B., Turk, M., and Höllerer, T. (2014b). World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 449–459. ACM.

- [Gericke et al., 2014] Gericke, L., Wenzel, M., and Meinel, C. (2014). Asynchronous understanding of creative sessions using archived collaboration artifacts. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 41–48. IEEE.
- [Gibson, 1979] Gibson, J. J. (1979). The ecological approach to visual perception.
- [Goebbels and Lalioti, 2001] Goebbels, G. and Lalioti, V. (2001). Co-presence and co-working in distributed collaborative virtual environments. In *Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation*, pages 109–114. ACM.
- [Goebbels et al., 2003] Goebbels, G., Lalioti, V., and Göbel, M. (2003). Design and evaluation of team work in distributed collaborative virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 231–238.
- [Goldschwendt et al., 2014] Goldschwendt, T., Anthes, C., Schubert, G., Kranzlmüller, D., and Petzold, F. (2014). The collaborative design platform protocol: a protocol for a mixed reality installation for improved incorporation of laypeople in architecture. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 239–240. ACM.
- [Greenhalgh and Benford, 1995] Greenhalgh, C. and Benford, S. (1995). MASSIVE: a collaborative virtual environment for teleconferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):239–261.
- [Gross et al., 2005] Gross, T., Stary, C., and Totter, A. (2005). User-centered awareness in computer-supported cooperative work-systems: Structured embedding of findings from social sciences. *International Journal of Human-Computer Interaction*, 18(3):323–360.
- [Gutwin and Greenberg, 1996] Gutwin, C. and Greenberg, S. (1996). Workspace awareness for groupware. In *Conference Companion on Human Factors in Computing Systems*, pages 208–209. ACM.
- [Gutwin and Greenberg, 2002] Gutwin, C. and Greenberg, S. (2002). A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3-4):411–446.
- [Han et al., 2016] Han, P.-H., Chen, K.-W., Hsieh, C.-H., Huang, Y.-J., and Hung, Y.-P. (2016). AR-Arm: Augmented Visualization for Guiding Arm Movement in the First-Person Perspective. In *Proceedings of the 7th Augmented Human International Conference 2016*, page 31. ACM.
- [Hinckley et al., 1998] Hinckley, K., Pausch, R., Proffitt, D., and Kassell, N. F. (1998). Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3):260–302.
- [Huang et al., 2011] Huang, W., Alem, L., and Albasri, J. (2011). HandsInAir: a wearable system for remote collaboration. *arXiv preprint arXiv:1112.1742*.
- [Huang et al., 2013a] Huang, W., Alem, L., Nepal, S., and Thilakanathan, D. (2013a). Supporting tele-assistance and tele-monitoring in safety-critical environments. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 539–542. ACM.
- [Huang et al., 2013b] Huang, W., Alem, L., and Tecchia, F. (2013b). HandsIn3d: augmenting the shared 3d visual space with unmediated hand gestures. In *SIGGRAPH Asia 2013 Emerging Technologies*, page 10. ACM.

- [Huang et al., 2013c] Huang, W., Alem, L., and Tecchia, F. (2013c). HandsIn3d: supporting remote guidance with immersive virtual environments. In *Human-Computer Interaction—INTERACT 2013*, pages 70–77. Springer.
- [Ibayashi et al., 2015] Ibayashi, H., Sugiura, Y., Sakamoto, D., Miyata, N., Tada, M., Okuma, T., Kurata, T., Mochimaru, M., and Igarashi, T. (2015). Dollhouse VR: a multi-view, multi-user collaborative design workspace with VR technology. In *SIGGRAPH Asia 2015 Emerging Technologies*, page 8. ACM.
- [Jacob et al., 2008] Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J. (2008). Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 201–210. ACM.
- [Kilteni et al., 2012] Kilteni, K., Groten, R., and Slater, M. (2012). The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387.
- [Koutek and Post, 2001] Koutek, M. and Post, F. H. (2001). *Spring-based manipulation tools for virtual environments*. Springer.
- [Lacoche et al., 2015a] Lacoche, J., Duval, T., Arnaldi, B., Maisel, E., and Royan, J. (2015a). Plasticity for 3d user interfaces: new models for devices and interaction techniques. In *EICS 2015: 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 28–33. ACM.
- [Lacoche et al., 2015b] Lacoche, J., Le Chénéchal, M., Chalmé, S., Royan, J., Duval, T., Gouranton, V., Maisel, r., and Arnaldi, B. (2015b). Dealing with frame cancellation for stereoscopic displays in 3d user interfaces. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 73–80. IEEE.
- [Larrodé et al., 2008] Larrodé, X., Chanclou, B., and Aguerreche, L. (2008). Openmask: an open-source platform for virtual reality. In *IEEE VR workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*.
- [Le Chenechal et al., 2015] Le Chenechal, M., Lacoche, J., Martin, C., and Royan, J. (2015). Laying out spaces with virtual reality. In *Virtual Reality (VR), 2015 IEEE*, pages 337–338. IEEE.
- [Le Chénéchal et al., 2014a] Le Chénéchal, M., Arnaldi, B., Duval, T., Gouranton, V., Royan, J., and others (2014a). From 3d Bimanual Toward Distant Collaborative Interaction Techniques: An Awareness Issue. In *2014 International Workshop on Collaborative Virtual Environments (3DCVE)*.
- [Le Chénéchal et al., 2014b] Le Chénéchal, M., Arnaldi, B., and Marchal, M. (2014b). Poster: Bimanual Design of Deformable Objects Thanks to the Multi-tool Visual Metaphor. In *3D User Interfaces*.
- [Lopez et al., 2013] Lopez, T., Gouranton, V., Nouviale, F., and Arnaldi, B. (2013). The ghost in the shell paradigm for virtual agents and users in collaborative virtual environments for training. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*, page 8. ACM.
- [Luna et al., 2012] Luna, A. S., Gouranton, V., and Arnaldi, B. (2012). Collaborative virtual environments for training: A unified interaction model for real humans and virtual humans. In *E-Learning and Games for Training, Education, Health and Sports*, pages 1–12. Springer.

- [Mallot, 1997] Mallot, H. A. (1997). Behavior-oriented approaches to cognition: Theoretical perspectives. *Theory in biosciences*, 116:196–220.
- [Marchal et al., 2011] Marchal, M., Pettr, J., and Lécuyer, A. (2011). Joyman: A human-scale joystick for navigating in virtual worlds. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 19–26. IEEE.
- [Margery et al., 1999] Margery, D., Arnaldi, B., and Plouzeau, N. (1999). A general framework for cooperative manipulation in virtual environments. In *Virtual Environments' 99*, pages 169–178. Springer.
- [Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329.
- [Mizutori et al., 2012] Mizutori, M., Hirota, K., and Ikei, Y. (2012). Skillful Manipulation of Virtual Objects Implementation of Juggling in a Virtual Environment. *Virtual Systems and Multimedia*, pages 79–86.
- [Mori et al., 2012] Mori, M., MacDorman, K. F., and Kageki, N. (2012). The uncanny valley [from the field]. *Robotics & Automation Magazine, IEEE*, 19(2):98–100.
- [Murayama et al., 2004] Murayama, J., Bougrila, L., Luo, Y., Akahane, K., Hasegawa, S., Hirsbrunner, B., and Sato, M. (2004). SPIDAR G&G: A two-handed haptic interface for bimanual VR interaction. In *Proceedings of EuroHaptics*, pages 138–146.
- [Nahon et al., 2015] Nahon, D., Subileau, G., and Capel, B. (2015). “Never Blind VR” enhancing the virtual reality headset experience with augmented virtuality. In *Virtual Reality (VR), 2015 IEEE*, pages 347–348. IEEE.
- [Naud et al., 2009] Naud, M., Ullah, S., Richard, P., Otmane, S., Mallem, M., et al. (2009). Effect of tactile feedback and viewpoint on task performance in a collaborative virtual environment. In *Joint virtual reality conf of EGVE-ICAT-EuroVR*.
- [Nguyen et al., 2013] Nguyen, T. T. H., Duval, T., and Fleury, C. (2013). Guiding Techniques for Collaborative Exploration in Multi-Scale Shared Virtual Environments. In *Proceedings of GRAPP International Conference on Computer Graphics Theory and Applications*.
- [Nguyen et al., 2012] Nguyen, T. T. H., Fleury, C., and Duval, T. (2012). Collaborative exploration in a multi-scale shared virtual environment. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 181–182.
- [Niewiadomski et al., 2011] Niewiadomski, R., Hyniewska, S. J., and Pelachaud, C. (2011). Constraint-based model for synthesis of multimodal sequential expressions of emotions. *Affective Computing, IEEE Transactions on*, 2(3):134–146.
- [Noh et al., 2015] Noh, S.-T., Yeo, H.-S., and Woo, W. (2015). An HMD-based Mixed Reality System for Avatar-Mediated Remote Collaboration with Bare-hand Interaction.
- [Ohlenburg et al., 2004] Ohlenburg, J., Herbst, I., Lindt, I., Fröhlich, T., and Broll, W. (2004). The MORGAN framework: enabling dynamic multi-user AR and VR projects. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 166–169.
- [Ortega and Coquillart, 2006] Ortega, M. and Coquillart, S. (2006). Prop-based haptic interaction with co-location and immersion: an automotive application. *arXiv preprint cs/0601025*.

BIBLIOGRAPHY

- [Pauchet et al., 2007] Pauchet, A., Coldefy, F., Lefebvre, L., Picard, S. L. D., Bouguet, A., Peron, L., Guerin, J., Corvaisier, D., and Collobert, M. (2007). Mutual awareness in collocated and distant collaborative tasks using shared interfaces. In *Human-Computer Interaction—INTERACT 2007*, pages 59–73. Springer.
- [Petit et al., 2009] Petit, B., Lesage, J.-D., Menier, C., Allard, J., Franco, J.-S., Raffin, B., Boyer, E., and Faure, F. (2009). Multicamera real-time 3d modeling for telepresence and remote collaboration. *International Journal of Digital Multimedia Broadcasting*, 2010.
- [Pinho et al., 2002] Pinho, M. S., Bowman, D. A., and Freitas, C. M. (2002). Cooperative object manipulation in immersive virtual environments: framework and techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 171–178.
- [Pinho et al., 2008] Pinho, M. S., Bowman, D. A., and Freitas, C. M. D. S. (2008). Cooperative object manipulation in collaborative virtual environments. *Journal of the Brazilian Computer Society*, 14(2):53–67.
- [Poston and Serra, 1994] Poston, T. and Serra, L. (1994). The virtual workbench: dextrous VR. In *Proceedings of the conference on Virtual reality software and technology*, pages 111–121. World Scientific Publishing Co., Inc.
- [Poupyrev et al., 1996] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). 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*, pages 79–80. ACM.
- [Rae et al., 2015] Rae, I., Venolia, G., Tang, J. C., and Molnar, D. (2015). A framework for understanding and designing telepresence. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1552–1566. ACM.
- [Rama and Bishop, 2006] Rama, J. and Bishop, J. (2006). A survey and comparison of CSCW groupware applications. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 198–205. South African Institute for Computer Scientists and Information Technologists.
- [Razzaque et al., 2001] Razzaque, S., Kohn, Z., and Whitton, M. C. (2001). Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, pages 105–106. Citeseer.
- [Riege et al., 2006] Riege, K., Holtkämper, T., Wesche, G., and Fröhlich, B. (2006). The bent pick ray: An extended pointing technique for multi-user interaction. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 62–65. IEEE.
- [Robert et al., 2013] Robert, K., Zhu, D., Huang, W., Alem, L., and Gedeon, T. (2013). MobileHelper: remote guiding using smart mobile devices, hand gestures and augmented reality. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, page 39. ACM.
- [Ruddle et al., 2002] Ruddle, R. A., Savage, J. C., and Jones, D. M. (2002). Symmetric and asymmetric action integration during cooperative object manipulation in virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(4):285–308.
- [Sallnäs et al., 2000] Sallnäs, E.-L., Rasmus-Gröhn, K., and Sjöström, C. (2000). Supporting presence in collaborative environments by haptic force feedback. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(4):461–476.

- [Salzmann et al., 2009] Salzmann, H., Jacobs, J., and Froehlich, B. (2009). Collaborative Interaction in Co-Located Two-User Scenarios. In *JVRC 2009 (Joint Virtual Reality Conference of EGVE - ICAT - EuroVR 2009)*, pages 85–92.
- [Sanchez-Vives and Slater, 2004] Sanchez-Vives, M. V. and Slater, M. (2004). From presence towards consciousness. In *8th Annual Conference for the Scientific Study of Consciousness*.
- [Slater and Wilbur, 1997] Slater, M. and Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments*, 6(6):603–616.
- [Sreng et al., 2006] Sreng, J., Lecuyer, A., and Megard, C. (2006). Using Visual Cues of Contact to Improve Interactive Manipulation of Virtual Objects in Industrial Assembly/Maintenance Simulations. *Visualization and Computer Graphics*, 12:1013–1020.
- [Stellmach and Dachsel, 2013] Stellmach, S. and Dachsel, R. (2013). Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 285–294. ACM.
- [Steptoe et al., 2008] Steptoe, W., Wolff, R., Murgia, A., Guimaraes, E., Rae, J., Sharkey, P., Roberts, D., and Steed, A. (2008). Eye-tracking for avatar eye-gaze and interactional analysis in immersive collaborative virtual environments. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 197–200.
- [Stoakley et al., 1995] Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co.
- [Strauss and Carey, 1992] Strauss, P. S. and Carey, R. (1992). An object-oriented 3d graphics toolkit. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 341–349. ACM.
- [Sukan et al., 2014] Sukan, M., Elvezio, C., Oda, O., Feiner, S., and Tversky, B. (2014). ParaFrustum: visualization techniques for guiding a user to a constrained set of viewing positions and orientations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 331–340. ACM.
- [Sutherland, 1965] Sutherland, I. E. (1965). The ultimate display. *Multimedia: From Wagner to virtual reality*.
- [Sutherland, 1968] Sutherland, I. E. (1968). A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM.
- [Tecchia et al., 2012] Tecchia, F., Alem, L., and Huang, W. (2012). 3d helping hands: a gesture based MR system for remote collaboration. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 323–328. ACM.
- [Thevenin and Coutaz, 1999] Thevenin, D. and Coutaz, J. (1999). Plasticity of user interfaces: Framework and research agenda. In *Proceedings of INTERACT*, volume 99, pages 110–117.
- [Tolani et al., 2000] Tolani, D., Goswami, A., and Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388.

- [Tonnis et al., 2005] Tonnis, M., Sandor, C., Lange, C., and Bubb, H. (2005). Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 56–59. IEEE Computer Society.
- [Towles et al., 2002] Towles, H., Chen, W.-C., Yang, R., Kum, S.-U., Kelshikar, H. F. N., Mulligan, J., Daniilidis, K., Fuchs, H., Hill, C. C., Mulligan, N. K. J., and others (2002). 3d tele-collaboration over internet2. In *In: International Workshop on Immersive Telepresence, Juan Les Pins*. Citeseer.
- [Usoh et al., 1999] Usoh, M., Arthur, K., Whitton, M. C., Bastos, R., Steed, A., Slater, M., and Brooks Jr, F. P. (1999). Walking> walking-in-place> flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364. ACM Press/Addison-Wesley Publishing Co.
- [Wartell et al., 2009] Wartell, Z., Houtgast, E., Pfeiffer, O., Shaw, C. D., Ribarsky, W., and Post, F. (2009). Interaction volume management in a multi-scale virtual environment. In *Advances in Information and Intelligent Systems*, pages 327–349. Springer.
- [Zahorik and Jenison, 1998] Zahorik, P. and Jenison, R. L. (1998). Presence as being-in-the-world. *Presence: Teleoperators and virtual environments*, 7(1):78–89.
- [Zillner et al., 2014] Zillner, J., Rhemann, C., Izadi, S., and Haller, M. (2014). 3d-board: a whole-body remote collaborative whiteboard. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 471–479. ACM.

Être capable de collaborer à distance avec d'autres personnes peut fournir de précieuses capacités pour effectuer des tâches qui ont besoin de plusieurs utilisateurs pour être accomplies. De plus, les technologies de Réalité Mixte (RM) sont des outils intéressants pour développer de nouveaux types d'applications offrant des interactions et des possibilités de perception plus naturelles comparées aux systèmes classiques. Dans cette thèse, nous proposons d'améliorer la collaboration distante en utilisant ces technologies de RM qui profitent de nos capacités naturelles à effectuer des tâches en environnements 3D. En particulier, nous nous concentrons sur les aspects asymétriques impliqués par ce type de collaboration : les rôles, le point de vue (PdV), les dispositifs et le niveau de virtualité de l'application de RM.

Premièrement, nous nous intéressons aux problèmes d'*awareness* et nous proposons un modèle générique capable de décrire précisément une application de RM collaborative en prenant en compte les potentielles dimensions asymétriques. Afin de traiter toutes ces dimensions, nous séparons notre modèle final en deux niveaux qui distinguent espaces réels et virtuels pour chaque utilisateur. Dans ce modèle, chaque utilisateur peut générer différents types d'entrées et recevoir des retours de significations différentes dans le but de maintenir leur propre *awareness* de l'Environnement Virtuel (EV) partagé. Puis, nous présentons une étude utilisateur exploratoire qui s'intéresse aux conséquences de l'asymétrie des PdVs et aux implications induites par la représentation des utilisateurs sur le niveau d'*awareness* des autres collaborateurs.

Deuxièmement, nous appliquons ces observations dans un contexte de guidage à distance qui implique un guide distant aidant un opérateur à réaliser une tâche de maintenance. Pour ce cas d'usage, nous proposons à l'expert d'utiliser une interface de Réalité Virtuelle (RV) pour aider l'opérateur au travers d'une interface de Réalité Augmentée (RA). Nous contribuons à ce domaine en améliorant les capacités de perception de l'environnement distant par l'expert et en proposant des interactions plus naturelles pour guider l'opérateur au travers d'indications non intrusives et intégrées à son environnement réel.

Finalement, nous abordons la tâche de co-manipulation qui est une situation encore plus sensible vis-à-vis de l'*awareness* en collaboration distante. Cette tâche requiert de viser une synchronisation parfaite entre les collaborateurs pour l'accomplir efficacement. Ainsi, le système doit fournir des retours appropriés pour maintenir un haut niveau d'*awareness*, spécialement concernant l'activité courante des autres. En particulier, nous proposons une technique de co-manipulation hybride, inspirée de notre cas d'utilisation précédent sur la guidage distant, qui mixe la manipulation d'objet virtuel et du PdV d'un autre utilisateur.

Being able to collaborate remotely with other people can provide valuable capabilities in performing tasks that require multiple users to be achieved. Moreover, Mixed Reality (MR) technologies are great tools to develop new kinds of applications with more natural interactions and perception abilities compared to classical desktop setups. In this thesis, we propose to improve remote collaboration using these MR technologies that take advantages of our natural skills to perform tasks in 3D environments. In particular, we focus on asymmetrical aspects involved by these kind of collaboration: roles, point of view (PoV), devices and level of virtuality of the MR application.

First, we focus on awareness issues and we propose a generic model able to accurately describe a collaborative MR application taking into account potential asymmetry dimensions. In order to address all these dimensions, we split our final model into two layers that separate real and virtual spaces for each user. In this model, each user can generate different kind of input and receive feedbacks with different meanings in order to maintain their own awareness of the shared Virtual Environment (VE). Then, we conduct an exploratory user study to explore the consequences of asymmetric PoVs and the involvement of users' representation in the level of awareness of others' collaborators.

Second, we apply our findings to a remote guiding context that implies a remote guide to help an operator in performing a maintenance task. For this use case, we propose to the expert to use a Virtual Reality (VR) interface in order to help the operator through an Augmented Reality (AR) interface. We contribute to this field by enhancing the expert's perceptual abilities of the remote workspace as well as by providing more natural interactions to guide the operator through not intrusive guiding cues integrated to the real world.

Last, we address an even more sensitive situation for awareness in remote collaboration that is virtual co-manipulation. It requires to target a perfect synchronization between collaborators in order to achieve the task efficiently. Thus, the system needs to provide appropriate feedbacks to maintain a high level of awareness, especially about what others are currently doing. In particular, we propose a hybrid co-manipulation technique, inspired from our previous remote guiding use case, that mixes virtual object and other's PoV manipulation in the same time.