



CIDER: Collaborative Interior Design in Extended Reality

Deborah Pintani
deborah.pintani@univr.it
University of Verona
Italy

Ariel Caputo
ariel.caputo@univr.it
University of Verona
Italy

Daniel Mendes
danielmendes@fe.up.pt
University of Porto
Portugal

Andrea Giachetti
andrea.giachetti@univr.it
University of Verona
Italy



Figure 1: Left: two subjects wearing the HoloLens 2 and using CIDER in our lab. Center: one of them places a virtual chair in the room and proposes the update of the global scene with a two-phase *commit* operation. Right: The other subject receives a notification and can see the proposed scene, accepting or refusing the global update. The internal views are directly recorded from the apps.

ABSTRACT

Despite significant efforts dedicated to exploring the potential applications of collaborative mixed reality, the focus of the existing works is mostly related to the creation of shared virtual/mixed environments resolving concurrent manipulation issues rather than supporting an effective collaboration strategy for the design procedure. For this reason, we present CIDER, a system for the collaborative editing of 3D augmented scenes allowing two or more users to manipulate the virtual scene elements independently and without unexpected changes. CIDER is based on the use of “layers” encapsulating the state of the environment with private layers that can be edited independently and a global one collaboratively updated with “commit” operations. Using this system, implemented for the HoloLens 2 headsets and supporting multiple users, we performed a user test on a realistic interior design task, evaluating the general usability and comparing two different approaches for the management of the atomic commit: forced (single-phase) and voting (requiring consensus), analyzing the effects of this choice on the collaborative behavior.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHIItaly 2023, September 20–22, 2023, Torino, Italy

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0806-0/23/09...\$15.00

<https://doi.org/10.1145/3605390.3605419>

CCS CONCEPTS

• **Human-centered computing** → **Collaborative and social computing systems and tools; Empirical studies in HCI; User studies; Mixed / augmented reality; Interaction design process and methods.**

KEYWORDS

HCI, Collaborative system, User study, Mixed reality

ACM Reference Format:

Deborah Pintani, Ariel Caputo, Daniel Mendes, and Andrea Giachetti. 2023. CIDER: Collaborative Interior Design in Extended Reality. In *15th Biannual Conference of the Italian SIGCHI Chapter (CHIItaly 2023)*, September 20–22, 2023, Torino, Italy. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3605390.3605419>

1 INTRODUCTION

Interactive tools based on eXtended reality (XR) are expected to replace existing Human-Computer interfaces [6] in a variety of professional and industrial sectors, not only for single-user tasks but also for collaborative ones. The interest in collaborative work in mixed environments has also grown commercially, indicating that the business world is also pushing in this direction. An example is Microsoft Mesh [4], which allows people in different physical locations to share the same virtual environment. While many works have been dedicated to the generic study of collaborative mixed reality [17, 28] and potential applications [32], few efforts have been dedicated to studying how users can perform concurrent editing and manipulation of virtual objects in co-located spaces, as noted in a recent survey [30].

Managing concurrent editing/manipulation of 3D objects in a collaborative mixed-reality environment is not straightforward and depends heavily on the user scenario. Most of the works describing collaborative prototypes for scene editing focus on properly managing access to shared objects, but, as pointed out in [25], many different needs should be considered for supporting collaboration in co-located AR experiences.

A reasonable scenario of use for applications like interior design in mixed reality assumes that different users may want to propose different variations of objects and scenes, compare the results, discuss them, and create optimally merged results without the risk of being surprised by unwanted changes of the scene under development.

In this paper, we propose CIDER, a solution designed for this specific scenario, based on *layers* and the smart management of their visualization for the different users. Users are co-located in the same environment and can add to it and edit different elements, working on local layers and only indirectly updating the shared layer through *commits*. We implemented two types of commits in our work: forced commit, allowing a single user to update the global layer without the explicit agreement of the others, and voting commit, requiring the explicit acceptance of the modifications.

The distributed application is lightweight, running on standalone HoloLens 2 devices without the need for external PC acting as servers. The HoloLens apps manage the local user layer and communicate with the other applications to maintain a shared synchronized scene representation (global layer). The apps also manage the local/global visualization switch and the forced or voting (two-phases) commit actions.

To limit the bandwidth required for the synchronization of the scenes, the applications rely on a shared dictionary of graphical objects pre-loaded in all devices so that only the ID of the instantiated objects and the modified properties (position, orientation, materials) are exchanged.

This layer-based approach allows users to work independently, even interacting with the same virtual objects without disturbing each other, and discuss and approve or reject the modifications made to the environment. This is also suitable to support asynchronous work, a case in which the voting commit can be useful for a user to become aware of the proposed changes when deciding upon those. The proposed approach has been tested on a two-users collaborative interior design task with 20 participants, comparing the behavior of the subject pairs with the different commit methods, and evaluating the general usability of the system.

Therefore, the main contributions of this paper are (1) a novel layer-based approach for concurrent manipulation of 3D virtual objects in a collaborative mixed-reality environment; (2) an implementation of the layer-based approach in a functional prototype with two different commit types; and (3) a user evaluation with 20 participants.

The repository containing the code of the application can be accessed at the following url:
<https://github.com/DebbyX3/CIDERHololensCollabManip>.

2 RELATED WORK

A huge amount of research work has been dedicated to the development of collaborative 3D scene design methods in extended reality.

Ong and Shen [24] proposed an early prototype of a mixed reality (MR) environment where users could visualize and interact with virtual products in a 3D physical space using intuitive virtual interfaces.

Coppens and Mens [16] realized a software bridge between Grasshopper, a popular tool for parametric architectural modeling, and the HTC Vive VR headset, analyzing the possible solutions to handle concurrency for multi-user settings but not implementing solutions to handle conflicts. They consider a basic locking strategy as the most reasonable option, provided that an appropriate unlocking mechanism takes into account unexpected disconnections. A simple locking strategy for concurrent immersive editing has been used, for example in [11] where a Web-based collaborative authoring tool for Virtual Reality is presented. An alternative solution for handling concurrent editing in a synchronized co-located mixed environment is the use of the operational transform paradigm [29]. This paradigm, popular in online text editors and other collaborative tools has been adapted to real-time collaborative 3D design systems in [7]. A similar approach was used for example by Grandi et al. [19] who proposed a handheld device-based AR collaborative editing environment with a fine-grained non-locking concurrent approach, allowing multiple users to apply independent transformations and then summing them without restrictions or weights. They experimented with this concurrence handling also in a test on asymmetric collaboration mixing VR and handheld-device-based AR.

Wieland et al. [33] evaluated different strategies to lock or compose degrees of freedom of manipulation controls between two users. In these papers, the collaboration strategies adopted by subject pairs were analyzed but the tasks were simple dockings and not simulations of real collaborative works. Actually, it is difficult to think that users may want to coordinate the collaborative placement of an item in the scene when it is effortless and it is not surprising that in [33] test subjects performed better while perceiving less workload when controlling separately the degree of freedoms (DOFs).

Linebarger and Kessler [22] considered some issues related to "surprises" (undo and intention) in synchronized collaborative editing due to conflicts and proposed solutions to handle them. The support for undoing action has also been proposed and discussed in [12].

As pointed out in [17], most of the works that support collaboration through mixed reality focus on synchronous collaboration scenarios in the time dimension.

However, the implementation of synchronized co-located editing with the management of the concurrency can be not sufficient to guarantee good usability in a collaborative design setting as the manipulation performed by a user can destroy the work planned by others. To enable the users to propose tentative design solutions to the collaborators without the risk of having them broken by them, a more reliable solution would be to allow asynchronous editing of proprietary copy interleaved with synchronization steps

with users able to switch between different augmented views. The idea of having multiple users with multiple augmentations and the ability of visualization changes has been proposed, for example in [27], even if not in an editing context.

In this paper, we propose a multi-layer solution (CIDER) where each user can edit a private layer and the update of the shared design is managed with commit operations like in the git versioning system. The system is able to manage effectively the synchronization of the work and the user-controlled visualization of different layers providing a satisfactory user experience and a smooth exchange of design ideas.

3 CIDER

3.1 The collaborative design system

We designed CIDER as a HoloLens 2 system usable by many collaborating subjects to augment collaboratively a real space with a set of virtual items using a natural interface allowing direct manipulation. The users can use specific interfaces to add objects to the scene, move, rotate, possibly resize them and change material parameters. The key aspect of CIDER is that users can work independently on a personal augmentations and see each other's changes in the shared space, but only if they want to.

The management of the independent editing, the visualization of the different changes, and the agreement on global design choices are based on *layers*.

A *layer*, also called a *scene*, is a container that encapsulates an environment state. Because we focus on 3D object manipulation, our layers contain a set of items with related positions, orientations, scales, and materials. Changes to objects in one layer do not affect the objects in another layer.

Since a collaborative environment features multiple users, it is possible to assign a layer to each one of them. In this case, each user has one *personal* or *local* layer.

To make the application collaborative, it is then necessary to add at least a *global* layer shared between all users. Users can edit this global layer only by uploading local changes to the global scene. In this case, it is necessary to implement a mechanism to request an agreement with all the users on the proposed changes.

3.2 Practical implementation

The application is built with Unity [31] and the Microsoft Mixed Reality Toolkit 2 (MRTK2) [5]. In this context, augmented scenes are characterized by instances of *GameObjects* representing menus, 3D elements, scenery, cameras, etc., that are defined in the real-world coordinates of the environment. To support shared real-world coordinate systems for all the HoloLens 2 devices we rely on a simple calibration procedure consisting of starting the devices with our apps with the same position and orientation.

UI components based on MRTK are shown to control the creation and deletion of objects and the changes of their materials, as well as for the visualization layer switch and commit management (Figure 2).

The management of the shared layer and the commit operations are based on a client-server architecture. Messages are exchanged by using the Transmission Control Protocol (TCP) and the programs running on the different headsets use traditional sockets to create



(a)



(b)

Figure 2: (a) "Add object" menu with the furniture catalog. (b) Menus to manage the layers' visualization and the commit operations (i.e. Force and Propose Commit) and different operations on the objects (i.e. Change Color and Duplicate)

a secure channel to communicate and create multiple threads to handle requests.

With these messages, the shared global layer is replicated and synchronized in all the applications, and the commit procedures are implemented. The amount of data transmitted is low thanks to the fact that the *GameObjects*' data are not sent across the network: only the operations needed to replicate the design given the furniture dictionary pre-loaded on all the running applications are transmitted.

Users receive a notification about commits or deletions. To view them, they press the corresponding button on the general menu. As evaluated by Rzayev et al. [26], a head-up display notification placement decreases the response time and the number of missed notifications. After pressing the button, a slate-like menu is displayed, showing the user a list of notifications and their details; pressing one of them takes the user to the global layer to see the change. A notification contains a short description, an image of the concerned object, its name, and its color.

Note that the notification button is disabled if there are no notifications. Moreover, the button turns red when there is at least one

notification, and a sound is played when receiving a new one, as suggested by Lazaro et al. [21].

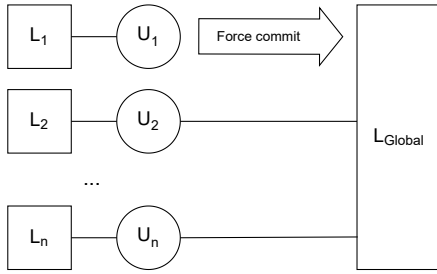
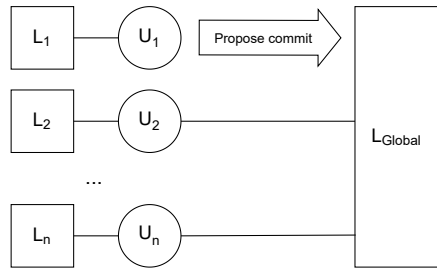
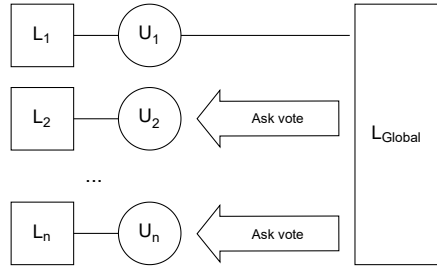


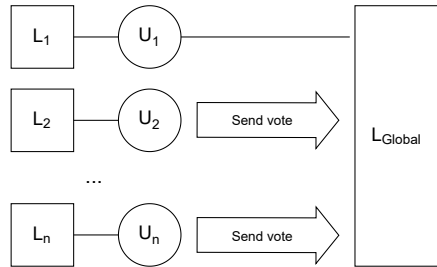
Figure 3: Forced commit diagram. $U_1, U_2 \dots U_n$ are the users. $L_1, L_2 \dots L_n$ are the local layers of each user. L_{Global} is the global layer. U_1 sends their forced commit to the global layer.



(a) U_1 sends their voting commit to the global layer



(b) Other users receive the commit and are asked to vote



(c) Receiving users send their vote

Figure 4: Voting commit diagram. $U_1, U_2 \dots U_n$ are the users. $L_1, L_2 \dots L_n$ are the local layers of each user. L_{Global} is the global layer. U_1 sends their voting commit to the global layer, and other users decide on the change.

3.3 Commit operations

To manage the agreement we decided to use a classical solution borrowed from the *Git version control system* [2].

A *Git commit* [1] is a snapshot of the changes in the source code of a software project recorded in the Git version control system. Git commits can be seen as a history of the project's progress and allow developers to revert to previous versions, compare changes, and collaborate with other contributors.

Similarly, in our work, a commit operation is the act of sending a personal change in the global layer to other users.

In our case, the mixed reality tool allows the user to create and modify virtual objects added to the real scene working in a "local" layer. A commit operation sends the ID, position, rotation, scale, and material properties of local layer objects to the global layer. An interesting feature of working with commit operations is that users are able to collaborate together from the beginning: commits are versatile and participants can adapt them to fit their workflow. In this regard, users may want to work in their personal layers and then discuss at the end by submitting the appropriate objects; alternatively, they may want to send their changes as soon as possible, so other users could see them.

In our work, we implemented and compared two types of commits management:

Forced commit When a user issues a *forced commit* from a local layer, the change is sent to the global layer without needing approval from other users and is immediately visible to them. Receiving users will get a notification that informs them about the new commit (see Figure 3 and Figure 5).

Voting or proposal commit When a user issues a *voting (or proposal) commit* from a local layer, the change is put to a vote: the other users receive a notification that invites them to see the change in the global layer and to accept or reject it. At this point, the object is in *pending state*, and it takes a yellow outline, so users more easily recognize it (Figure 6). If the modification is accepted, the new object's state is confirmed and committed to the global layer, sending a notification to all users. Otherwise, if the change is rejected, the object will revert to its global state before the commit. This type of commit borrows the first part of the Two-Phase Commit protocol [8], the *commit-request phase (or voting phase)* (Figure 4).

Note that, for both methods, only the global layer is affected by changes: the personal ones do not change, even if a voting commit is rejected. Commit operations concern both object creations and changes. If an item was never committed before (therefore it does not exist in the global layer) then the first commit adds it to the shared layer. Otherwise, if an item already exists in the global layer, then issuing a commit would change its state in the shared layer, overwriting its previous global state.

Commits are implemented via messages holding the following object's information: GUID (GameObject unique ID), Transform (position, rotation and scale), PrefabName (containing the name of the Prefab), MaterialName (representing the name of the material). Messages are serialized and sent to the network.

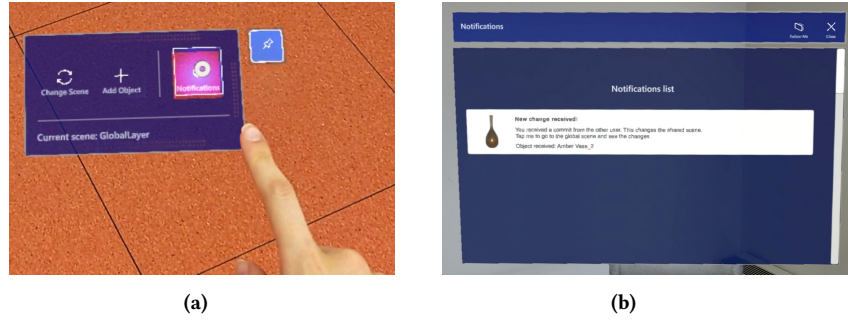


Figure 5: Receiving a Forced Commit. (a) User pushes the "Notifications" button. The red color signals that there are unread notifications. (b) Notification panel. The notification informs the reception of a new forced commit. No more actions are needed from the user.

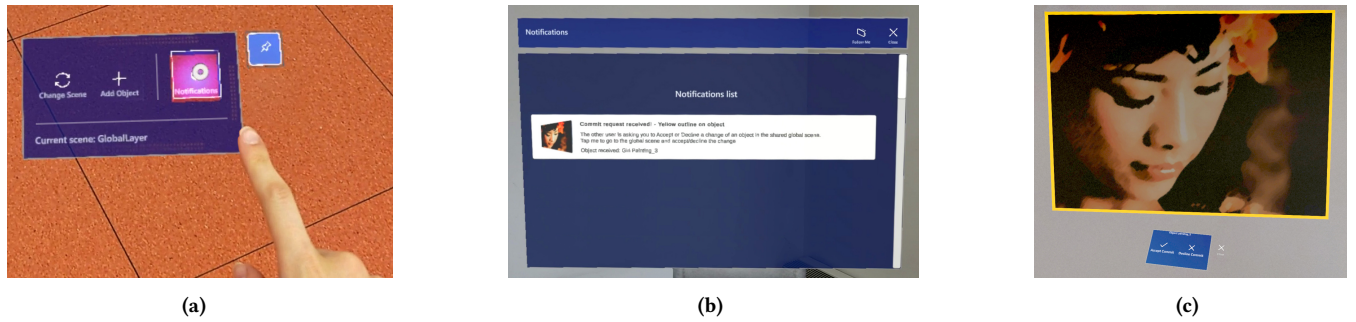


Figure 6: Receiving a Voting Commit. (a) User pushes the "Notifications" button. The red color signals that there are unread notifications. (b) Notification panel. The notification informs the reception of a new voting commit, which has to be accepted or rejected. (c) The object involved in the received commit request takes a yellow outline. Below, the menu to accept or reject the change.

3.4 Memento pattern

In the application, an object can acquire multiple states, one for each layer where it appears into. For example, an item can be green in the global layer, but blue in the local, or alternatively it can just exist as a green object in the global layer. When switching between layers, it is essential for the app to store objects' states before showing new ones, especially if an item belongs to more than one scene. Regarding the last case and recalling the last example, it is important to save the green color of the object in the global layer when changing to the local one, so that when it is time to switch back to the global scene, the item will restore its green color. Vice-versa, when changing to the local layer, the application must re-assign the blue color to the object after saving the previous global green color and, consequently, save the local blue color when switching back to the global scene. A suitable design pattern to model this operation is the *Memento* pattern. The Memento pattern [18] is a behavioral design pattern that allows the saving and restoring of an object's state. This pattern stores a snapshot of the item's properties and is used to restore an object to its previous state, even after it has been modified. It is specifically built to not violate encapsulation and to externalize the internal state of an object.

In this pattern, three objects are involved: the *originator* (the object

whose state needs to be saved), the *memento* (an object that stores the originator's state), and the *caretaker* (an object that manages the memento's lifecycle). The originator creates and uses mementos to save and restore its state, while the caretaker is responsible for storing and retrieving mementos. In our app, a variant of the Memento pattern is developed to better suit the specific use cases that this work addresses. The caretaker actually stores multiple dictionaries that contain a pair <GUID, Memento> for each object in the scene that each dictionary refers to. For example, a dedicated list called *LocalListMementos* stores the mentioned pairs for each object that belongs to the local scene. Naturally, an object may have several states: it can belong to the local layer but also be in a pending commit state due to a voting commit, and waiting to be accepted by the other user. It is then necessary to have multiple dictionaries to effectively model and store the various objects' states.

The save functionality is implemented by the class controlling the Game Object behaviour, which creates a Memento object that stores a snapshot of its current internal state.

The restore functionality is implemented by the caretaker, which retrieves a Memento object that was previously stored and passes it back to the object controller which then uses the Memento object to restore its internal state. The restore operation is performed at each layer switch, to reload the appropriate objects' state for the changed

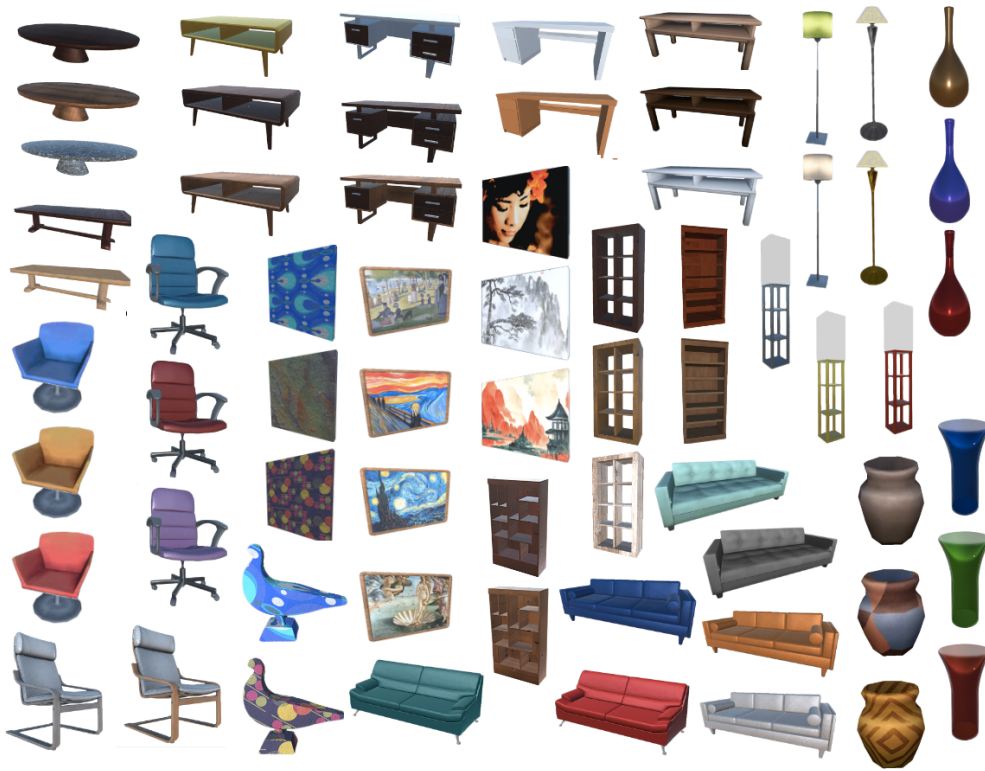


Figure 7: The catalog of furniture items available for the design task

scene. It's important to note that the originator object itself doesn't know how to restore its state: this responsibility is delegated to the memento object. This allows the originator object to maintain encapsulation, since it doesn't need to expose its internal state to other objects in the system.

4 USER STUDY DESIGN

For a first evaluation of CIDER, we designed a simple user study, asking subjects' pairs to use it to furnish a room together in Augmented Reality. The goal of the study was to evaluate the general usability of the system and compare the two commit procedures using a within-subject experimental design. The study was approved by the Person Research Approval Committee (CARP) of the University of Verona.

We selected 20 participants, 4 females and 16 males between the age of 19 and 27, all computer science students but two. The subjects did not know each other and were divided into 10 pairs chosen at random.

Subject pairs were asked to furnish a room together in Augmented Reality, in 10 minutes, using each of the specific commit types.

They were instructed on how to use CIDER with a specific tutorial and had a fixed time to familiarize themselves with the system and to test editing and commit operations.

During the task, the subjects could furnish the specific rooms using a furniture catalog within the app (shown in Figure 7) to pick the furniture of their liking. In particular, to test both commit types,

participants furnished a living room using a commit method, and an office with the other (example results shown in Figure 8).

To prevent biases, the room's order and commit types are alternated with a Latin square design.

Each room at the end of the task, in the global scene, should include at least four pieces of furniture: for the living room, at least a coffee table, a painting, a sofa, and a shelf, and for the office, at least an office chair, a lamp, a shelf, and a desk. Other elements could have been added optionally to the scene.

The dictionaries of available furniture elements included at least three different models for each category, each one available in two or more different materials.

For each task performed with the two commit modes, we collected the objective and subjective metrics described in the following.

4.1 Objective metrics

During the tasks, we recorded a set of system variables collected for each user at each time frame: active layer (local vs global), gaze direction, gaze origin, gaze targeted object GUID (if available) head position, head direction, head orientation, and currently manipulated object GUID (if available). We also recorded the type, object involved, timestamp, and outcomes of all the commit operations issued. To evaluate the collaborative behavior, we also measured the amount of time spent talking.

4.2 Questionnaire

After the completion of the tasks with each commit system, we asked the subjects to fill out a questionnaire.

The questionnaire was divided into three parts. The first one focused on the quality of the collaborative work and included questions adapted from Jakobsen and Hornbæk [20], which evaluated collaborative work using a multi-touch wall display, and questions from Biocca et al. [9], which specifically offers a method to measure social presence in online environments. The answers to those questions describe participants' perceptions of how they collaborated during the task. The second cluster of questions was directly taken from the System Usability Scale (SUS) by Brooke [10], which is a flexible method to assess the overall usability of a system. The third group focused on the emotional state of participants with questions regarding their mood while using the system. These questions are inspired by Lockner and Bonnardel [23], which explores the relationship between interface design and user emotions, and proposes a method for measuring the emotional effects of interface design. This section also contains questions from Cho and Kim [14], which measures user emotion and experience in interaction with space. It is worth mentioning Chin et al. [13] who developed a Questionnaire for User Interface Satisfaction (QUIS): although these questions were not directly included in the questionnaire of this work, QUIS inspired some of them.

All the questions rate the agreement with specific sentences on a Likert scale from 1 (totally disagree) to 5 (totally agree). The full set of questions is reported in Table 1.



(a) Living room



(b) Office

Figure 8: Examples of final augmented environments created by participants of the CIDER evaluation test, as seen from the application.

5 STUDY OUTCOMES

All the subjects' pairs completed the tasks of creating the augmented rooms as required. Examples of the results are shown in Figure 8.

Looking at the logged data, we can see that the two commit methods did not result in significant differences in the average completion time or in the time spent talking (that we may assume linked to the degree of collaboration), but some qualitative differences, related to the behavior of specific pairs were recorded. The violin plots in Figure 9a show the distribution of the times required to complete the tasks by the pairs using forced and voting commit. It is possible to see that even if the differences in the averages (represented by the green dashed lines) are not statistically significant, it is possible to see some qualitative differences in the distributions, the variability of the times is clearly higher for forced commits and only with this method we had three pairs completing the task in a relevantly shorter time. This indicates that if a quick completion is required, the force commit option could be preferred.

Figure 9b, shows the distributions of the percentages of time spent talking while performing the tasks, divided into forced and voting commit. Here too we cannot see significant differences in the average, but a clear difference in the distributions: when performing voting commit tasks there is no middle ground: users either talk a lot or very little. This may indicate that for some subjects the voting commit procedure may replace the discussion on the modifications proposed, and this is not similarly feasible with the forced commit.

We measured a statistically significant difference in the average percentages of time spent by the subjects in the global layer visualization mode with the two commit modes (24, 75% with the Forced commit and 34.91% with the Voting one). These values may be related to the amount of collaborative work performed during the tasks, as they measure the time spent analyzing the shared design. These percentages are normally distributed (Shapiro $s = 0.91$, $p = 0.08$) and a paired t-test comparing the means revealed a significant difference ($t = 4.06$, $p < 0.001$). The trend can be seen clearly in the slope graph of Figure 10a.

Another statistically significant difference measured was between the average number of changes applied to the global layer by each pair of subjects for the two methods, respectively 10.9 for the forced and 6.7 for the voting method (as shown in the slope graph in Figure 10b). For the forced method the number of forced commits and item deletions were counted, while the number of accepted commits and accepted item deletions were counted for the voting method. The Shapiro test revealed the data were not normally distributed ($p < 0.05$) and the non-parametric Mann-Whitney U test shows a significant difference ($U = 85.0$, $p = 0.009$). These results hint at two different strategies of resolving conflicts between the users. The lower number of changes to the global layer for the voting method, indicates that the users are taking the chance of going through the voting process to discuss and agree on the outcome of each proposed change to the global layer, while the higher number of changes for the forced method, suggests that the users settle for a certain outcome after a series of forced commits. We also tested for differences in the number of commits, with averages of 10.6 and 9.6 respectively for the forced and voting methods, but

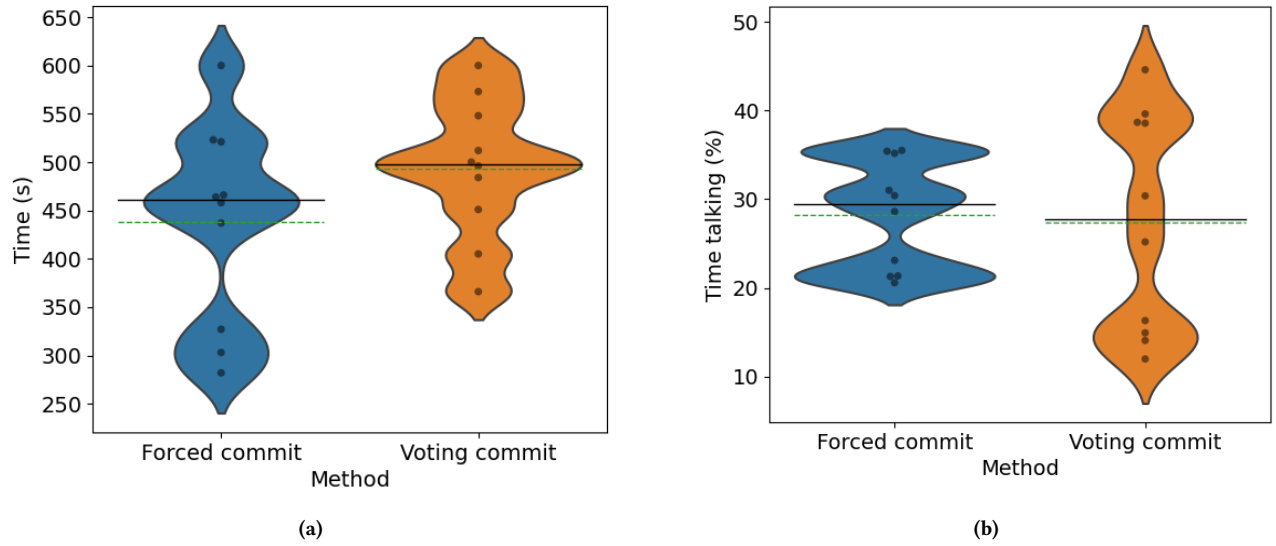


Figure 9: (a) Distribution of task completion times of the 10 pairs with the forced (left, blue) and voting (right, orange) commit modes. (b) Distribution of the percentage of time spent talking by the 10 pairs with the forced (left, blue) and voting (right, orange) commit modes.

could not find a significant difference (t-test with $p = 0.591$). This somewhat supports the interpretation of the previous result as it shows there's no difference in the number of times a user wanted to apply a change. However, equivalence between the two averages could not be proven as a TOST equivalence test returned a non significant result ($p = 0.5$).

Qualitatively we observed that one of the pairs organized the collaboration in a completely different way, as the subjects worked independently and compared different proposals when almost complete, while the others organized the works in small steps with frequent interactions. For this group, we recorded a large difference in the talking time between the two commit methods (31% with force commit and 12% with voting commit).

5.1 Questionnaire

The average ratings (with standard deviations) for the questions are reported in Table 1, grouped by commit type. It is evident, looking at the figures, that no significant differences are found across the commit types for all the questions. The scores are rather high for all three aspects investigated (collaboration, usability, mood), showing that our implementation is ready for practical use and both the commit modes are easy to understand. It is worth noting that there are a few outliers in the scores, for example, related to the feeling about the contribution of the partner, but in a similarly low number for both the commit modes.

The average SUS score for forced commit tasks is 81.38, while for voting commit tasks is 80.38. Both values are exceeding the 85% percentile, according to [3], and this means that we can consider the system usability very good in both versions.

Figure 11a shows a plot of SUS scores. The distribution is more homogeneous for the voting commit, while for the forced one the scores are mostly in the 80-90 range. A few issues may have limited

the usability for a few subjects with the voting commit, for example, looking at Figure 11b showing the distributions of the answers to the question "I felt happy performing the task" it appears that three subjects were less happy to perform the task while using the voting commit system (score of 3). It can be assumed that voting commits may enforce more arguments, but this aspect needs clearly further investigation.

The questionnaire also included open questions and the subjects could comment on other issues related to the task. Only one subject considered the management of the voting commit a bit complex, while another one had some problems with the direct manipulation with hand gestures provided by MRTK.

6 DISCUSSION

The idea behind CIDER is to enable a co-located collaboration on an augmented scene allowing independent user work on private layers and synchronized updates with commits, a usual procedure in software development, but not found in current collaborative 3D design tools or research prototypes. There are many use cases for which this solution can give relevant advantages, for example, to allow non-expert subjects to collaborate in the design of scenes, to help experts to show some potential solutions in a particular environment, or to get ideas from laypeople. In [15] the authors demonstrated that an Immersive Virtual Environment (IVE) facilitated instrument can allow laypeople to actively take part as designers in the early stage of an urban design process. We believe that our system can similarly help the collaborative work of professionals as well the exchange of ideas between designers and customers.

We believe that the mechanisms implemented in CIDER help to meet the requirements of an effective collaboration [25], as it allows the subjects to

Question 1 (= strongly disagree) to 5 (=totally agree):	Forced commit		Voting commit	
	Mean	SD	Mean	SD
<i>Collaborative work</i>				
I worked closely together with my partner	4,50	0,89	4,45	0,89
I was fully aware of what my partner was doing	4,35	0,88	4,15	0,93
I contributed a lot to the final design	4,50	0,76	4,50	0,76
The group work was very effective	4,70	0,57	4,70	0,57
I am very satisfied of my work in the team	4,90	0,31	4,70	0,57
My partner did not help me very much	1,30	0,80	1,40	0,94
The other understood what I meant	4,85	0,37	4,80	0,41
I understood what the other meant	4,85	0,37	4,90	0,31
<i>System Usability Score (SUS)</i>				
I think that I would like to use this system frequently	4,30	0,86	4,25	0,79
I found the system unnecessarily complex	1,30	0,47	1,60	0,88
I thought the system was easy to use	4,35	0,59	4,15	0,81
I think that I would need the support of a technical person to be able to use this system	2,05	0,94	2,30	1,03
I found the various functions in this system were well integrated	4,35	0,59	4,40	0,60
I thought there was too much inconsistency in this system	1,75	0,64	1,35	0,49
I would imagine that most people would learn to use this system very quickly	3,90	0,85	3,85	0,88
I found the system very cumbersome to use	1,95	1,10	2,10	0,85
I felt very confident using the system	4,25	0,72	4,10	0,97
I needed to learn a lot of things before I could get going with this system	1,55	0,69	1,25	0,44
<i>Mood</i>				
I felt happy performing the task	4,80	0,41	4,65	0,59
I felt to be more dominant than the other user regarding performing decisions	2,05	1,00	2,15	0,93
I felt the other user to be more dominant than me regarding performing decisions	2,10	1,07	2,35	1,04
I found it hard to think properly or concentrate	1,20	0,41	1,45	0,60
I did enjoy the task	4,95	0,22	4,80	0,41
It was hard for me to make up my mind	1,40	0,60	1,65	0,99
I felt calm performing the task	4,55	0,51	4,40	0,75
I felt angry performing the task	1,25	0,44	1,25	0,44
I felt anxious performing the task	1,25	0,72	1,25	0,44
I was bored during the task	1,10	0,31	1,30	0,73
I feel satisfied about the final result	4,80	0,41	4,70	0,47

Table 1: Questionnaire administered to user study participants divided by type of commit. SD is the standard deviation

- fully exploit the communication channels provided by the optical see-through augmented reality with direct talk and gestural communication
- *have a private space within a collaborative space* [25], being able to test augmentations without being influenced by other users [22]
- *be aware of others' attention and activities* [25], thanks to the notification mechanisms and the visualization solutions
- *see the same virtual content and have a smooth networked experience* [25] thanks to the limited size of the messages transmitted to maintain the shared layer
- safely edit the same objects without concurrency issues
- recover earlier versions of the project thanks to the memento pattern.

The outcomes of our first evaluation of the system confirmed that CIDER is easy to use and immediately understood by non-expert subjects, but are inconclusive about the optimal design of the commit execution.

However, the analysis of the subjects' behavior revealed that the system supports smoothly different collaboration styles with similar, good usability. Also, the fact that the voting procedure that could better support different collaborative styles and user scenarios, and may have been perceived as laborious and time-consuming is not impacting the performances and the overall usability of the system can be seen as a really interesting result.

The experimental evaluation presented here is clearly limited. The number of participants was low and the discussions on different design proposals have been less than expected, as most of the time the elements proposed by the partners have been only adjusted, and only a few times deleted.

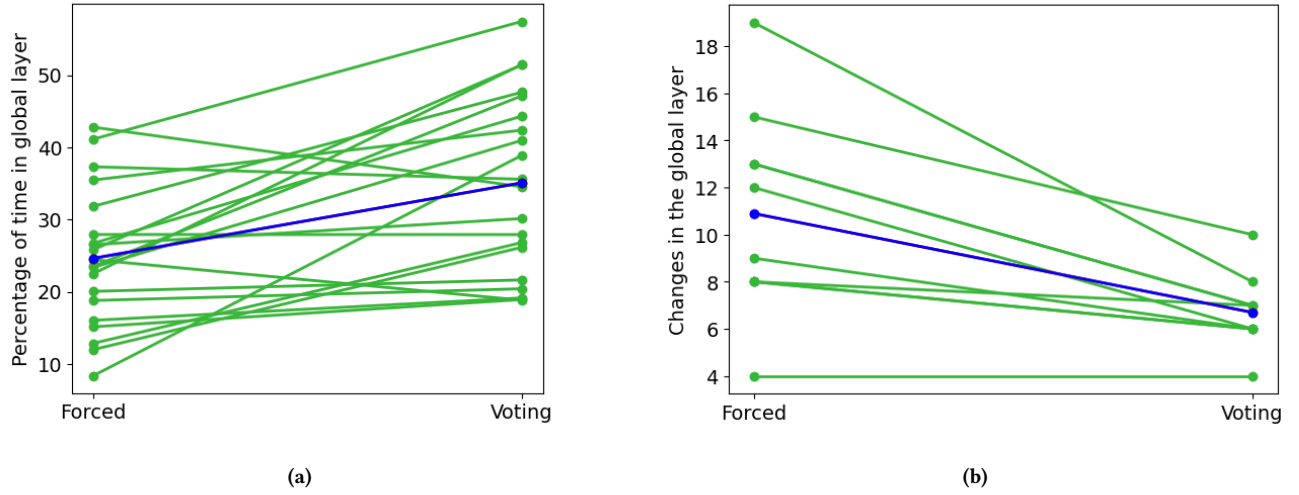


Figure 10: a) Slope graph representing the time spent in the global layer for the same subject when working with the Forced and Voting commit modes. The blue line represents the average value. (b) Slope graph representing the number of changes applied to the global layer (including deleting items) by each pair with the two commit modes. The blue line represents the average value.

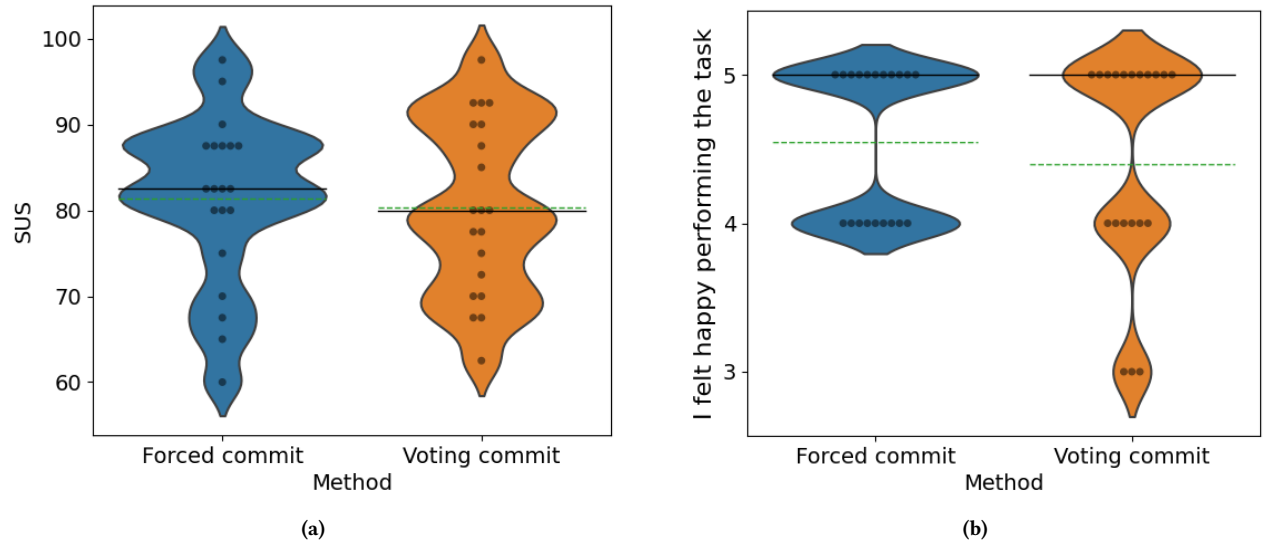


Figure 11: (a) Violin/swarm plots representing the distribution of SUS scores for forced and voting commit. (b) Violin/swarm plots representing the distribution of the answers to the "I felt happy performing the task" question.

The fact that the subjects were not interior design experts and the novelty of the interface made them focus on the AR experience rather than on the task, and they were not sufficiently motivated to support their design solutions, criticizing the other ones proposed. The different properties of the commit methods can make a difference only when there are different ideas and arguments between the users, but this happens in a limited number of cases with our experimental design. We could in principle try to force discussions with a different task, but trying to induce artificially some roles could have resulted in non-realistic behaviors.

Future tests on the system will be performed on more realistic use scenarios, that can go beyond the spatially and temporally co-located collaboration of two or more users editing a scene.

In fact, as it is designed like a versioning system, CIDER can be effectively exploited for asynchronous work. One or more designers/end users could propose solutions for a specific room to be evaluated by others later with the goal of realizing a collaborative design.

The system could be easily extended to support remote users immersed in a virtual reproduction of the local scene perceived

by the co-located users exploiting the XR headsets. It would be sufficient to send a 3D model or an implicit representation of the room, reconstructed from the HoloLens 2 sensors to a VR app to have the remote user work collaborating in the same way immersed in a "duplicated" reality [34].

7 CONCLUSIONS

In this paper, we described the design and implementation of CIDER, a system allowing the collaborative creation of an interior design adding virtual elements to the real environment. The system enables several users wearing HoloLens HMDs to work synchronously or asynchronously on private layers being enabled to commit editing operations on a shared global layer.

We also conducted a first experimental evaluation of the system with non-expert subjects comparing their behavior while adopting two different commit strategies. While the validation performed is limited by the simplified task and the non-expert subjects since we introduced a novel collaboration workflow to participants that was well received and the fact that the more laborious voting commit did not negatively affect the collaboration, we can highlight the potential of our approach. We believe that this work might contribute to better mixed-reality collaborative tools in the future, going in a different direction than the current approaches which function is based on locking objects and imposing changes.

ACKNOWLEDGMENTS

This study was partially carried out within the PNRR research activities of the consortium iNEST (Interconnected North-East Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS00000043).

REFERENCES

- [1] 2023. *git commit* documentation. <https://git-scm.com/docs/git-commit> Online; accessed March 2023.
- [2] 2023. *git Version Control System*. <https://git-scm.com> Online; accessed March 2023.
- [3] 2023. Interpret SUS score. <https://measuringu.com/interpret-sus-score/> Online; accessed March 2023.
- [4] 2023. Microsoft Mesh. <https://www.microsoft.com/en-us/mesh>
- [5] 2023. Mixed Reality Toolkit 2 - MRTK2. <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/> Online; accessed March 2023.
- [6] Michael Abrash. 2021. Creating the future: augmented reality, the next human-machine interface. In *2021 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 1–2.
- [7] Agustina, Chengzheng Sun, and Dong Xu. 2012. Operational Transformation for Dependency Conflict Resolution in Real-Time Collaborative 3D Design Systems. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) (CSCW '12). Association for Computing Machinery, New York, NY, USA, 1401–1410. <https://doi.org/10.1145/2145204.2145412>
- [8] Yousef Al-houmaily and George Samaras. 2009. *Two-Phase Commit*. 3204–3209. https://doi.org/10.1007/978-1-4899-7993-3_713-2
- [9] Frank Biocca, Chad Harms, and Jennifer Gregg. 2001. The Networked Minds Measure of Social Presence: Pilot Test of the Factor Structure and Concurrent Validity. *4th annual International Workshop on Presence, Philadelphia* (01 2001).
- [10] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).
- [11] Nicola Capece, Ugo Erra, Giuseppe Losasso, and Francesco D'Andrea. 2019. Design and implementation of a web-based collaborative authoring tool for the virtual reality. In *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 603–610.
- [12] Yuan Cheng, Fazhi He, Xiantao Cai, and Dejun Zhang. 2013. A group Undo/Redo method in 3D collaborative modeling systems with performance evaluation. *Journal of Network and Computer Applications* 36, 6 (2013), 1512–1522.
- [13] John P. Chin, Virginia A. Diehl, and Kent L. Norman. 1988. Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., USA) (CHI '88). Association for Computing Machinery, New York, NY, USA, 213–218. <https://doi.org/10.1145/57167.57203>
- [14] Myung Cho and Mi jeong Kim. 2017. Measurement of User Emotion and Experience in Interaction with Space. *Journal of Asian Architecture and Building Engineering* 16 (01 2017), 99–106. <https://doi.org/10.3130/jaabe.16.99>
- [15] Shuva Chowdhury and Marc Aurel Schnabel. 2019. Laypeople's Collaborative Immersive Virtual Reality Design Discourse in Neighborhood Design. *Frontiers in Robotics and AI* 6 (2019), 97.
- [16] Adrien Coppens and Tom Mens. 2018. Towards collaborative immersive environments for parametric modelling. In *International Conference on Cooperative Design, Visualization and Engineering*. Springer, 304–307.
- [17] Barrett Ens, Joel Lanir, Anthony Tang, Scott Bateman, Gun Lee, Thammathip Piumsomboon, and Mark Billingham. 2019. Revisiting collaboration through mixed reality: The evolution of groupware. *International Journal of Human-Computer Studies* 131 (2019), 81–98. <https://doi.org/10.1016/j.ijhcs.2019.05.011>
- [18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education. <https://books.google.it/books?id=6oHuKQe3TjQC>
- [19] Jerônimo G Grandi, Henrique G Debarba, Iago Bemdt, Luciana Nedel, and Anderson Maciel. 2018. Design and assessment of a collaborative 3D interaction technique for handheld augmented reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 49–56.
- [20] Mikkel R. Jakobsen and Kasper Hornbæk. 2014. Up Close and Personal: Collaborative Work on a High-Resolution Multitouch Wall Display. *ACM Trans. Comput.-Hum. Interact.* 21, 2, Article 11 (feb 2014), 34 pages. <https://doi.org/10.1145/2576099>
- [21] May Jorella Lazaro, Sungho Kim, Jaeyong Lee, Jaemin Chun, and Myung-Hwan Yun. 2021. Interaction Modalities for Notification Signals in Augmented Reality. 470–477. <https://doi.org/10.1145/3462244.3479898>
- [22] John M. Linebarger and G. Drew Kessler. 2004. Concurrency Control Mechanisms for Closely Coupled Collaboration in Multithreaded Peer-to-Peer Virtual Environments. *Presence* 13, 3 (2004), 296–314. <https://doi.org/10.1162/1054746041422316>
- [23] Damien Lockner and Nathalie Bonnardel. 2014. Emotion and Interface Design How to measure interface design emotional effect?
- [24] S.K. Ong and Y. Shen. 2009. A mixed reality environment for collaborative product design and development. *CIRP Annals* 58, 1 (2009), 139–142. <https://doi.org/10.1016/j.cirp.2009.03.020>
- [25] Iulian Radu, Tugce Joy, Yiran Bowman, Ian Bott, and Bertrand Schneider. 2021. A Survey of Needs and Features for Augmented Reality Collaborations in Collocated Spaces. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–21.
- [26] Rufat Rzaev, Sven Mayer, Christian Krauter, and Niels Henze. 2019. Notification in VR: The Effect of Notification Placement, Task and Environment. 199–211. <https://doi.org/10.1145/3311350.3347190>
- [27] Dieter Schmalstieg, Gerhard Reitmayr, and Gerd Hesina. 2003. Distributed applications for collaborative three-dimensional workspaces. *Presence* 12, 1 (2003), 52–67.
- [28] Mickael Sereno, Xiyao Wang, Lonni Besançon, Michael J Mcguffin, and Tobias Isenberg. 2022. Collaborative Work in Augmented Reality: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 28, 6 (2022), 2530–2549.
- [29] David Sun, Steven Xia, Chengzheng Sun, and David Chen. 2004. Operational transformation for collaborative word processing. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 437–446.
- [30] Toqeer Ali Syed, Muhammad Shoaib Siddiqui, Hurria Binte Abdullah, Salman Jan, Abdallah Namoun, Ali Alzahrani, Adnan Nadeem, and Ahmad B. Alkhodre. 2023. In-Depth Review of Augmented Reality: Tracking Technologies, Development Tools, AR Displays, Collaborative AR, and Security Concerns. *Sensors* 23, 1 (2023). <https://doi.org/10.3390/s23010146>
- [31] Unity Technologies. 2023. Unity Website. <https://unity.com> Online; accessed March 2023.
- [32] Zhuo Wang, Xiaoliang Bai, Shusheng Zhang, Mark Billingham, Weiping He, Peng Wang, Weiqi Lan, Haitao Min, and Yu Chen. 2022. A comprehensive review of augmented reality-based instruction in manual assembly, training and repair. *Robotics and Computer-Integrated Manufacturing* 78 (2022), 102407.
- [33] Jonathan Wieland, Johannes Zagermann, Jens Müller, and Harald Reiterer. 2021. Separation, Composition, or Hybrid?—Comparing Collaborative 3D Object Manipulation Techniques for Handheld Augmented Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 403–412.
- [34] Kevin Yu, Ulrich Eck, Frieder Pankratz, Marc Lazarovici, Dirk Wilhelm, and Nassir Navab. 2022. Duplicated Reality for Co-located Augmented Reality Collaboration. *IEEE Transactions on Visualization and Computer Graphics* 28, 5 (2022), 2190–2200. <https://doi.org/10.1109/TVCG.2022.3150520>