

Cooperative Object Manipulation in Collaborative Virtual Environments

Marcio S. Pinho¹, Doug A. Bowman² and Carla M. Dal Sasso Freitas³

¹Faculdade de Informática
PUCRS
Av. Ipiranga, 6681
Phone: +55 (44) 32635874 (FAX)
CEP 13081-970 - Porto Alegre - RS - BRAZIL
marcio.pinho@pucrs.br

²Department of Computer Science
Virginia Polytechnic Institute and State University
P.O. Box 6101 – Zip 13083-852 – Blacksburg – Virginia - USA
bowman@vt.edu

³Instituto de Informática
Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 - CEP 91.501-970 - Porto Alegre – RS - BRAZIL
carla@inf.ufrgs.br

Received 8 June 2008; accepted 20 June 2008

Abstract

Cooperative manipulation refers to the simultaneous manipulation of a virtual object by multiple users in an immersive virtual environment (VE). In this work, we present techniques for cooperative manipulation based on existing single-user techniques. We discuss methods of combining simultaneous user actions, based on the separation of degrees of freedom between two users, and the awareness tools used to provide the necessary knowledge of the partner activities during the cooperative interaction process. We also present a framework for supporting the development of cooperative manipulation techniques, which are based on rules for combining single user interaction techniques. Finally, we report an evaluation of cooperative manipulation scenarios, the results indicating that, in certain situations, cooperative manipulation is more efficient and usable than single-user manipulation.

Keywords: Cooperative interaction; Collaborative interaction; Virtual environments; Interaction techniques; VR experiments.

1. INTRODUCTION AND MOTIVATION

Research on cooperative manipulation of objects in immersive virtual environments (VEs) is relevant in many areas, such as simulation and training, as well as in data exploration [24]. Cooperative manipulation refers to the simultaneous manipulation of a virtual object by multiple users in a VE. In simulation and training, simultaneous manipulation of objects in VEs can be used to mimic some aspects of real-world tasks. For example, in situations like product and equipment design, assembly tasks or emergency training, even when the users are not co-located in space, cooperative manipulation may provide more realistic interaction. In data exploration, cooperative manipulation is an important tool to enhance the interaction process, by moving it from being one-sided (“I do this, while you watch”) to being truly cooperative, increasing insight exchange and reducing the time for task completion.

The need for cooperative manipulation arises from the fact that some object manipulation tasks in VEs are difficult for a single user to perform with typical 3D interaction techniques. One example is when a user, using a ray-casting

technique, has to place an object far from its current position, which can be difficult if the user does not see all the surroundings of the aimed position. Another example is the manipulation of a large object without changing to a World-in-miniature (WIM) paradigm. In both cases, two users can perform the task more easily because they can both advise each other while performing cooperative, synchronized movements they are not able to perform alone.

Of course, some problems of these types can be addressed without cooperative manipulation: a single user could employ two-handed interaction to manipulate large objects in WIM environments, or a user could be allowed to simply advise his collaborator, both acting at separate times on the shared object. Although research on two-handed interaction has evolved over the years, bimanual interaction is usually applied to model the manipulation a single user would perform in a real world situation (see [7] and [13]). Bimanual tasks would be unnatural in many scenarios where the WIM paradigm is not the best solution, such as in cooperative structural design. For the situation where isolated, synchronized actions are employed, existing architectures are sufficient to support the collaboration. If, however, it is necessary or desired that more than one user be able to act at the same time on the same object, new interaction techniques and support tools need to be developed.

Our work is focused on how to support cooperative interaction and how to modify existing interaction techniques to fulfill the needs of cooperative tasks. To support the development of such techniques we have built a framework that allows us to explore various ways to separate degrees of freedom and to provide awareness for two users performing a cooperative manipulation task. We also aim at providing a seamless and natural transition between a single-user and a collaborative task, without any sort of explicit command or discontinuity in the interactive process, thus preserving the sense of immersion in the VE.

We base the design of our interaction techniques on rules that define how to combine and extend single-user interaction techniques in order to allow cooperative manipulation. We noticed that the state-of-the-art in single-user object manipulation was in the so-called “magic” [5] interaction techniques, based on the concept of mapping the user’s motions in some novel way to the degrees of freedom (DOFs) of the object. In these cases, the “magic” is used not to replace entirely the natural interaction but to augment the user’s capabilities. Usually this approach involves designing interaction techniques around cultural clichés and metaphors, such as the flying carpet or magic wand metaphors [6]. Classical examples of interaction metaphors used to create interaction techniques for virtual environments are Ray-Casting [18], 3D Magic Lenses [29], World-in-miniature [28].

We also noticed that cooperative manipulation techniques were mostly based on “natural” interaction (for example, simulation of the forces that each user applies to a virtual object). Simply combining these two approaches would create a discontinuity when users transitioned from single-user to cooperative manipulation.

Based on this observation, our work strives to show that “magic” interaction techniques can also be used efficiently in cooperative manipulation, in the sense that each user could control a certain subset of the DOFs associated with an object, thus minimizing the control that is required when a single user has to deal with multiple degrees of freedom at the same time to perform a task.

Considering the broader area of Computer Support for Collaborative Work (CSCW) and, specifically, groupware research as characterized by Wainer and Barsottini [30], our work reports the *design and evaluation* of an architecture for combining interaction techniques in order to support truly cooperative work regarding manipulation of objects in a virtual environment.

The paper is organized as follows. Next section characterizes cooperative manipulation of objects based on the difficulties arising in different situations. Section 3 surveys the existing approaches to provide for cooperative manipulation while section 4 presents our approach. In section 5, we briefly describe the software architecture which supports the development of cooperative manipulation techniques based on single-user techniques while in sections 6 and 7 we present and discuss experiments conducted for evaluation purposes. Finally, in section 8 we draw our conclusions and point out some future research.

This paper considerably extends our previous work [21] which described the support architecture for cooperative interaction and presented some preliminary findings on this topic. Here we are concerned with a deep description of the issues regarding cooperative interaction and present a detailed analysis of new cooperative techniques and task scenarios.

2. CHARACTERIZATION OF COOPERATIVE MANIPULATION

The original motivation for this work lies in the fact that certain VE manipulation tasks are more difficult when performed by a single user. These difficulties can be related to the interaction technique being used or to the task itself. In this section we discuss these difficulties.

2.1. DIFFICULTIES RELATED TO THE INTERACTION TECHNIQUE IN USE

Interaction techniques facilitate or complicate object manipulation in various ways. When using the ray-casting

technique [19], for instance, some rotations are difficult because this technique does not afford rotation around the vertical axis. To perform a task that involves this kind of rotation, a technique like HOMER [3] would certainly be a better option, because while it keeps the ray for object selection, it allows the user to easily rotate the selected object centered on its own coordinate system. Figure 1 shows that using a ray casting technique, the rotation of the user's pointer will move the object as if it were attached to the pointer. On the other hand, using HOMER, the same rotation will be mapped to the object's rotation around its own axis. Users of HOMER, however, have difficulty with certain types of object translation, because the ray orientation depends on the positions of the user's body and hand.

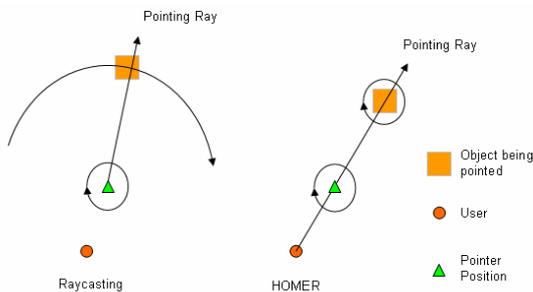


Figure 1: Different mappings of the same user's actions

Another possible solution is to allow the user to navigate to a better position to perform the rotation. However, if the environment presents too many obstacles, like walls or other objects, the navigation may be difficult also. Moreover, navigation introduces an additional level of complexity to the interactive process because the constant switches between navigation and manipulation increase the cognitive load and break the natural pace of operation [19] [23].

We could also consider the option of having multiple views of the environment, allowing the user to switch instantly between two or more positions during the interaction, but these would also affect the sense of immersion and could produce disorientation.

Another example of the limitations of interaction techniques is presented in Figure 2, in which a user U1 needs to move an object O from position A to position B without touching the obstacles. If the available interaction technique is the direct manipulation with the hand, then U1 will have to navigate to move the object. This will create additional difficulties, for U1 will have to release and grab the object many times in order to avoid the obstacles along the way. If HOMER, ray-casting or Go-Go [22] is used, navigation will not be necessary, but the translation parallel to the horizontal axis will not be easy to accomplish.

In this situation, a second user U2 next to point B may be able to help by sliding the object along the ray.

2.2. DIFFICULTIES RELATED TO THE TASK TO BE PERFORMED

Another motivation for cooperative manipulation comes from situations where the position, size or shape of the object introduces difficulties in its positioning and orientation. An example is when the object is distant from users or just partially visible. If a user has to place an object inside a shelf that is directly in front of him, as in Figure 3a, both horizontal and vertical positioning are simple. However, this user cannot easily determine the depth of the object for proper placement. A second user, shown in Figure 3b, can more easily perceive the depth and so help the first user to perform the task.

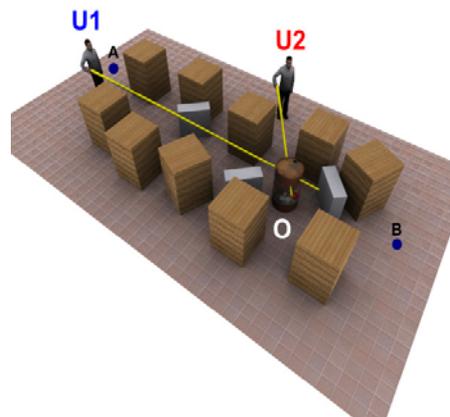


Figure 2: Object translation with obstacles

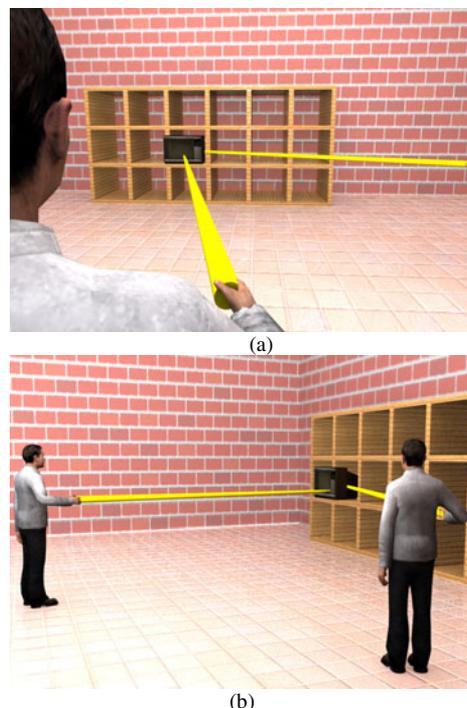


Figure 3: User without the notion of distance between an object and its final position

Another example involves the movement of a relatively large object through small spaces, such as moving of a couch through a door (Figure 4). Regardless the interaction technique being used, this task can become rather complex, especially if on the other side of the door there is an obstacle that can not be seen by the user who is manipulating the object. A second user, placed on the other side of the wall, can help the first one in accomplishing this task. This situation is similar to the “piano movers task” studied by Ruddle and colleagues [23][27].

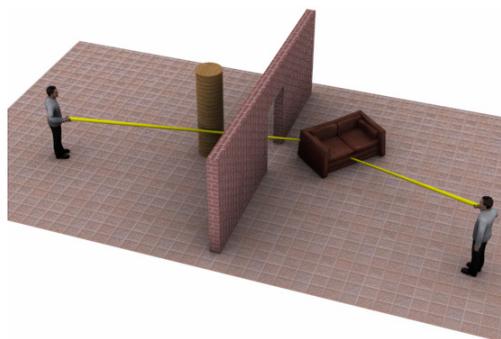
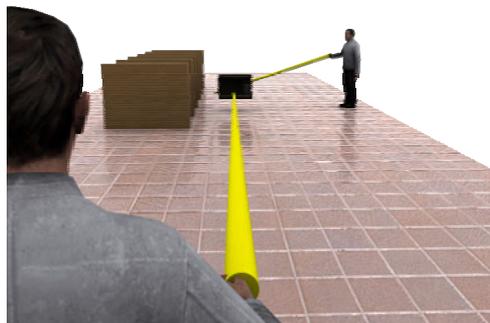


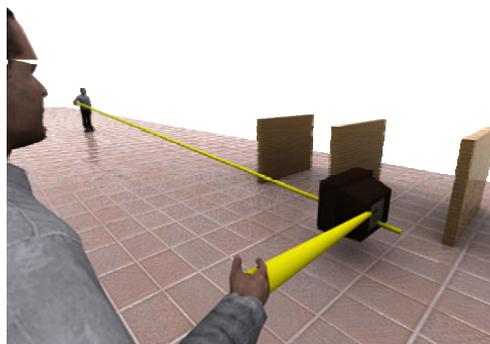
Figure 4: Movement of objects between obstacles

The manipulation of remote (distant) objects, which has been the focus of some prior work [3][20], is another example where cooperative manipulation can make the task’s execution easier. In Figure 5, for example, if user U1 has to place a computer between the panels he will have difficulties because he cannot clearly see the target location. In this case, a second user (U2) placed in a different position can help user U1 to find the proper position of the object.

these systems is schematically demonstrated in Figure 7, where it can be observed that the commands generated by each user are combined, producing a new command to be applied to each local copy of the object.



User U1's view



User U2's view

Figure 5: Manipulation of distant objects

3. EXISTENT APPROACHES FOR SUPPORTING COOPERATIVE MANIPULATION

In most of the known collaborative virtual environments (CVEs) like NPSNET [16], MASSIVE [12], Bamboo [31], DIVE [10], RAVEL [14], AVOCADO [11] and Urbi et Orbi [9], the simultaneous manipulation of the same object by multiple users is avoided. In these systems, the object receives a single command that is chosen from among many simultaneous commands applied to the object. Figure 6 shows a diagram modeling non-cooperative manipulation. Through an interaction technique a user executes an action that is converted into a command to be sent to the object. A second user performs a different action on the same object. The commands are received by a selector that decides which one must be applied to the object.

True cooperative manipulation has been the focus of a few research efforts. Most of these systems used force feedback devices so that each user senses the actions of the other [1] [26]. The manipulation process used in

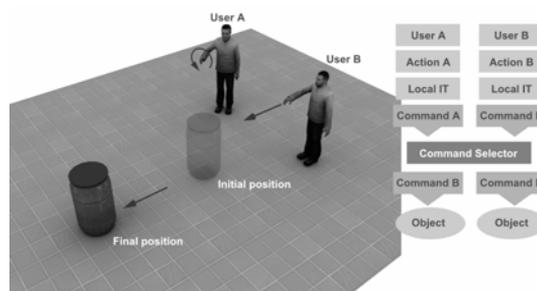


Figure 6: Command selection architecture: from the commands issued by the users, a single one is selected and applied to all the copies seen by different users

Margery [17] presents an architecture to allow cooperative manipulation without the use of force feedback devices. The system is restricted to a non-immersive environment, and the commands that can be applied to objects are vectors defining direction, orientation, intensity and the point of application of a

force upon the object. Thus, Margery’s work is based on the simulation of real-world cooperative manipulation.

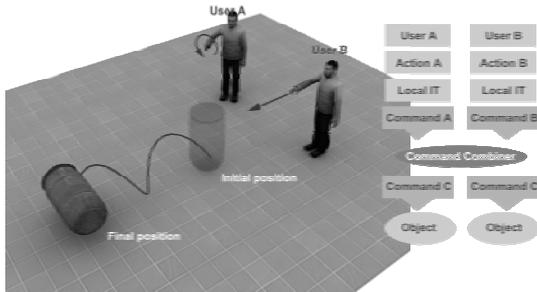


Figure 7: Command combination architecture: the commands issued by the users are combined and a new command is created and applied to all copies seen by different users

Earlier research by Ruddle and colleagues [23][24] presented the concept of rules of interaction to support symmetric and asymmetric manipulation. They are especially concerned with the maneuvering of large objects in cluttered VEs. In symmetric manipulation an object can only be moved if the users manipulate it in exactly the same way, while in asymmetric manipulation the object moves according to some aggregate of all users’ manipulation. Their work, however, uses only natural manipulation, and does not consider “magic” interaction techniques.

In a subsequent work, Ruddle et al. [25] separated collaborative tasks into two levels of control. The high level control activities correspond to those tasks that require attention, planning and mental effort by the users to be executed. One example of these activities is to define the general direction and speed of travel. On the other hand, the low level control activities are quasi-autonomous activities that, once learned, are easy and quickly executed by the users with no conscious control. Walking and grabbing objects are examples of such activities. The automation of these activities is possible, according to the authors, due to the flexibility with which one can move and the high-detail sensory feedback one obtains from real objects. In VEs, however, the feedback is of lower fidelity (and often completely missing), causing these tasks to require a high level of cognitive control. To overcome these problems their work proposes to encapsulate, in the VE software, knowledge about the tasks the user performs. So, tasks like grabbing objects and avoiding obstacles are automatically executed, decreasing the cognitive load for the task. Tests were run with a real user interacting with an autonomous virtual human. The results show that this approach can significantly reduce the time for task completion.

Recently, Duval et al. [8] presented a cooperative manipulation technique based on ‘crushing points’, considering the size and the geometry of the object. Two crushing points

define a “skewer” across the object. According to the authors, the users feel like they are pulling the object by a virtual cord. The proposed technique uses only the user’s hand position to apply translations and orientation changes to the object. The only problem reported for this technique is that rotation around the axis of the skewer is not allowed. To do so, the users have to release the object and select new crushing points, or new controls (like buttons or 6 DOF trackers) must be added to the interaction process.

4. COOPERATIVE MANIPULATION

The works surveyed in the previous section deals only with direct hand object manipulation, ignoring the very useful “magic” interaction techniques. Taking that into account, we developed a novel interaction model in which two users act in a simultaneous way upon the same object. Our approach combines individual interaction techniques instead of simply combining force vectors, creating an extension of the single-user interaction techniques commonly used in immersive VEs.

4.1. SINGLE-USER MANIPULATION TECHNIQUES

An interaction technique defines a mapping between user’s actions and their effects on an object. In our work, to model an interaction technique, we use Bowman’s methodology [4], which divides manipulation into four distinct components: selection, attachment, position and release. Each component has a corresponding phase in the interaction. Table 1 shows the meaning of each component.

Table 1: Components of an interactive manipulation technique

Component	Description
Selection	Specifies the method used for indicating an object to be manipulated.
Attachment	Specifies what happens when the object is captured by the user (or linked to its pointer)
Position	Specifies how the user’s and the pointer’s movements affect the object’s movement
Release	Specifies what happens when the object is released by the user

The use of this decomposition facilitates the combination of interaction techniques because each component can be treated separately. It is worth mentioning that all the interaction between user and object in the VE is done through a pointer controlled by the user. The shape and function of this pointer depend on the individual interaction technique.

4.2. COMBINING INTERACTION TECHNIQUES

Based on the decomposition presented above, we define a set of rules that defines how to combine and extend single-user interaction techniques in order to allow cooperative manipulation. Thus, our cooperative

interaction includes:

- How to combine actions in each phase of the interactive process when users are collaborating, and
- What kind of awareness must be generated in order to help the users understand each component of the cooperative interaction.

We also consider the following issues in the design of our cooperative manipulation techniques:

- Evolution: Building cooperative techniques as natural extensions of existing single-user techniques, in order to take advantage of prior user knowledge,
- Transition: Moving between a single-user and a collaborative task in a seamless and natural way without any sort of explicit command or discontinuity in the interactive process, preserving the sense of immersion in the virtual environment, and
- Code reuse: The subdivision of the interaction technique into well-defined components, allowing the designer to modify only the necessary parts of the single-user techniques to define a new cooperative technique.

In the next sections we examine how to combine each component of two or more interaction techniques to support simultaneous interaction.

4.2.1 . COMBINATION OF THE SELECTION COMPONENT

In the selection phase the collaborative activity begins. From the interaction technique point of view, the way in which an object is selected does not change whether the interaction is individual or collaborative. This is because simultaneous manipulation does not take place until both users confirm the selection of the same object. The way one user selects an object does not depend on whether or not his partner is manipulating the object. This property helps in the learning of the cooperative technique, because if the user already knows how to select an object with his individual interaction technique, he will not need to learn anything else to select the object for cooperative work.

4.2.2. COMBINATION OF THE ATTACHMENT COMPONENT

During the attachment of an object to a user's pointer, it is first necessary to verify whether the object is being manipulated by another user. If it is not, single-user manipulation proceeds normally. A message should also be sent to the partner, letting him know that one of the objects has just been attached to another user.

If another user is already manipulating the object, it is necessary to verify which DOFs can be controlled by each one, and set up functions to map each user's actions to the object based on these DOFs.

4.2.3. COMBINATION OF THE POSITION COMPONENT

The process of positioning an object in a simultaneous manipulation is based on the pointer's movement. If the local control system receives information related to the partner's pointer at each rendering cycle, it can locally perform the proper interpretation of this information, based on the cooperative manipulation rules, and apply the resulting commands to the object. This strategy eliminates the need for sending explicit commands related to the simultaneous manipulation situation through the network.

4.2.4. COMBINATION OF THE RELEASE COMPONENT

When an object is released, we should determine whether or not there is another user manipulating the object. If there is not, the functions that map from pointer's movements to commands should be disabled and a message sent to the partner. From then on the interactive process goes back to the selection phase.

If a second user is manipulating the same object, he must be notified that his partner has released the object. In our system, upon receiving the notification message he automatically releases the object. This way both users return to the selection phase and can restart the interactive process. In the first versions of our system, when a user received a message saying that his partner had released the object, he started to manipulate the object individually. This was not effective because the mapping rules of the movements were unexpectedly and involuntarily modified. From then on the user was able to control all the DOFs that were allowed by his individual interaction technique, without any notice whatsoever or possibility for controlling/reversing the situation. This almost always caused an undesired modification in the object placement just obtained with the cooperative interaction. After some trials, we noticed that the users began to synchronize the release of the object, trying to avoid undesired modifications in the object's position and orientation. The automatic double release allows a smooth transition from a collaborative to an individual activity.

4.3. AWARENESS FOR COOPERATIVE MANIPULATION

In this section we present the features related to awareness generation in each phase of the collaborative interaction process.

4.3.1. AWARENESS IN THE SELECTION PHASE

While the user chooses the object he wants to manipulate, it is essential that his partner know what is going on. This awareness will serve as a support to the interactive process. The pointer representation is used to allow a user to visualize what his partner is pointing to, and also to enable him to indicate an object he wants to manipulate or reference. Using such pointers, dialogues based on dietic references [2] like the one in Figure 8, can take place in a CVE.

User 1:	No, this is not the one! Please, get the object that is in front of this one I am pointing at.
User 2:	Which one? This or this one?

Figure 8: Dialogue supported by pointing

We can also use the shape or color of the pointer to allow a user to predict the interactive capabilities of his partner. In our system, when a user points to an object, that object takes on the color of the user's pointer.

During selection it is also necessary to provide awareness of two more states that can occur in collaborative environments. When one user has already attached an object to his pointer and, at the same time, the partner points to the object, we display the object using a third, different color. When both users, simultaneously point to the same object we use a less saturated version of this color.

4.3.2. AWARENESS IN THE ATTACHMENT PHASE

The attachment phase is a transition between the state in which the object is free and the state in which it is controlled by one or two users. During this transition two events occur, one related to the object and another related to the user's pointer. The object is highlighted somehow to signal that it is attached to a particular pointer. The pointer shape is also modified according to the interaction technique that is being used.

In our system, if only one user performs the attachment, the object goes back to its original color. In our first implementation, the object kept the pointer's color with a slightly greater intensity. Often, however, the users did not realize that the attachment had taken place, and they frequently complained that the original color would help in choosing the position/orientation of the object.

In a collaborative situation, when one user attaches to an object that is already attached to another user, the pointers for both users should be modified so that they represent which DOFs can be manipulated by each of them.

In our system, three different representations were used for three types of DOFs: rotation, translation and sliding along a pointing ray, also called "reeling". To demonstrate that a user can translate an object, the pointer turns into a set of one to three arrows, each of them representing an axis along which the object can be moved. Figure 9 shows some examples of pointers for translation. On the left, we can see the configuration of a pointer that allows a user to move the object only horizontally (plane XZ), and on the right another pointer that tells the user he can only move the object along Y axis.

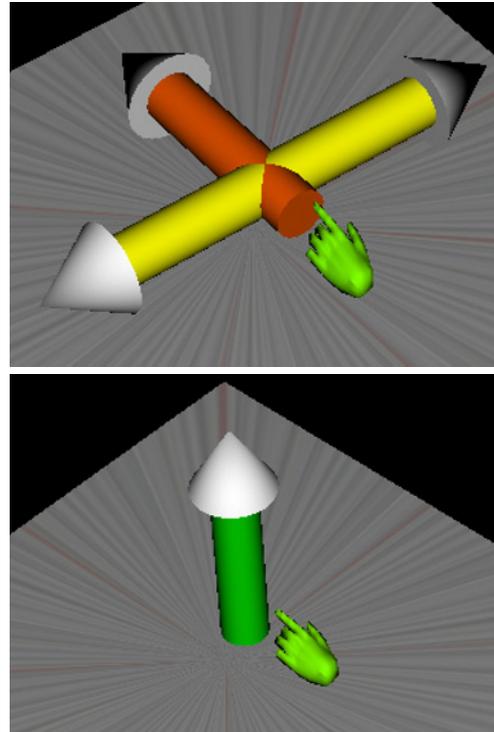


Figure 9: Examples of translation pointers

For rotation, the pointer turns into small disks that define the axes around which the user can rotate the object. Figure 10 shows two examples of pointers: the one on the left shows that the user can only rotate the object around Z-axis, while the one on the right indicates that all rotations are possible.

In order to provide to the user the notion he can slide an object along a ray, a longer arrow was introduced in the pointer representation. This arrow can be displayed in the same color as his own pointer or his partner's color. In the first case, the color indicates the user can slide the object along his own pointer and, in the second case, that it is possible for him to slide the object along his partner's pointer. Figure 11 shows this awareness tool combined with translation and rotation pointers.

It is possible to do any combination of the three types of pointers, indicating all the DOFs that a user can control for an object.

4.3.3. AWARENESS IN THE POSITIONING PHASE

During the cooperative positioning phase, the object is manipulated according to the rules of the cooperative interaction technique, without any special awareness information.

4.3.4. AWARENESS IN THE RELEASE PHASE

From the awareness point of view, the releasing phase reconfigures the pointers back to their original state, according to the individual interaction technique rules.

4.3.5. REPRESENTATION OF THE USER’S BODY

The graphical representation of the user’s body in a CVE supports body-relative positioning. This feature allows partners to point to elements in a natural way based on common conventions used during collaboration in real environments. We might hear, for example, the sentence: “Take the lamp that is in the box to your left and place it between us, within the reach of my right hand.”

An avatar should also represent the user’s head position and orientation. This allows other users to understand the gaze direction of the partner.

Although such avatars may not be necessary for accomplishing the tasks, they improve the sense of immersion and collaboration between partners.

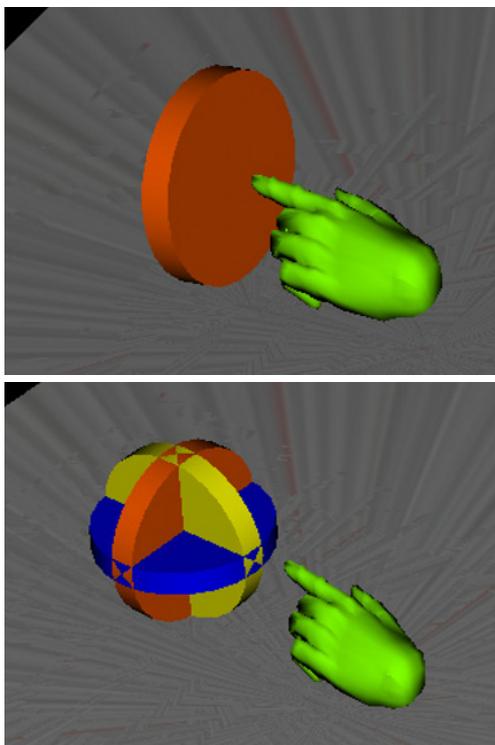


Figure 10: Examples of rotation pointers

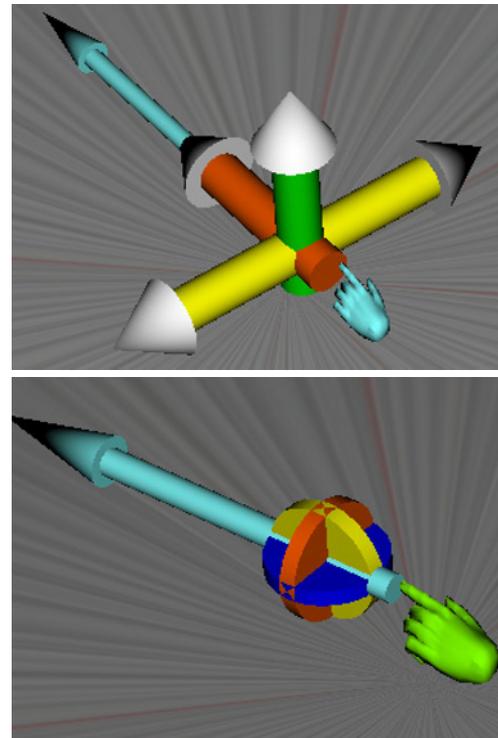


Figure 11: Examples of combined pointers

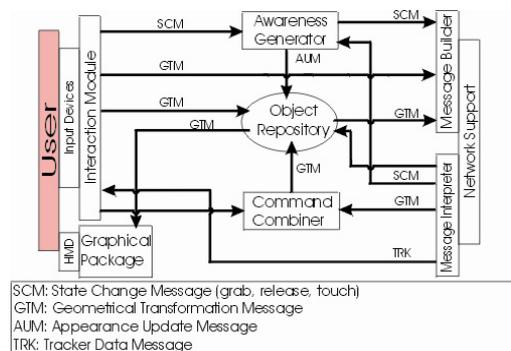


Figure 12: Architecture for cooperative manipulation based on the collaboration

5. SOFTWARE ARCHITECTURE

In this section, we provide a brief overview of our system’s architecture. A more detailed description was presented elsewhere [21].

In order to support the methodology presented in the previous section we have developed a software architecture (Figure 12) that provides:

- The independence of the individual techniques;
- The exchange of messages among partners;
- The generation of awareness and,
- The combination of commands.

The **Interaction Module** is responsible for mapping the pointer movements and commands generated by a user into transformations to be applied to the virtual object. This mapping is based on the individual (or cooperative) interaction technique’s specification. The **Input Devices** module reads the pointer movements.

Implemented as a single module, the **Graphical System** and the **Object Repository** generate the image of the VE that is displayed to the user. In this work, the VE is composed of a set of geometric objects rendered using the Simple Virtual Environment (SVE) library [14]. The geometric data that define the VE are

replicated on each machine taking part in the collaboration, in order to reduce network traffic.

The **Command Combiner** is activated when a cooperative interaction is established and it receives messages about the user’s pointer position from the **Interaction Module**, and messages about the position of the partner’s pointer from the Message Interpreter. Based on the cooperation rules that it implements, this module takes the received messages and, in every rendering cycle, selects which DOFs will be used from each user to update the position of the object that is being cooperatively manipulated. After generating a new transformation, the **Command Combiner** sends a message to the **Object Repository** in order to update the object position.

The **Awareness Generator** is the module responsible for updating the colors of the objects when the pointers are touching them, and it is also responsible for modifying the pointers’ shapes whenever a cooperative manipulation situation is established or finished. This module receives messages from the **Interaction Module** which originate from the interpretation of the local user’s movements and also from the **Message Interpreter**.

The **Message Interpreter** receives the messages coming from the partner and decides to which local module they should be sent.

Table 2 shows the set of existing messages, their meaning and the module to which they are sent by the **Message Interpreter**.

The **Message Builder** processes the messages received from the local modules and sends them to the partner. The **Network Support** module is responsible for sending and receiving the messages between the partners. This module is built on top of the TCP/IP protocol in order to ensure the proper synchronization between the environments and the consistency of the data that travel through the nodes.

Table 2: Messages received by the Messages’ Interpreter

Message	Local destination module(s)	Meaning
UPDATE Position	Command Combiner/ Object Repository	The object was moved by the partner
TRACKER data	Interaction Module	Tracker device data
GRAB event	Command Combiner	An object was attached by the user
RELEASE event	Command Combiner	An object was released by the user
TOUCH event	Awareness Generator	An object was touched by the user
UNTOUCH event	Awareness Generator	An object is not being touched anymore

6. EXPERIMENTS

In order to evaluate the use of our techniques for performing cooperative manipulation tasks in CVEs, we developed three VEs that allow two users to perform both cooperative and non-cooperative tasks. Our goal was to find specific situations where cooperative manipulation can lead to easier and more efficient task execution.

Each VE evaluates one combination of two single-user techniques. To choose the interaction techniques, both single-user and collaborative pilot studies were conducted. In these studies, “expert” VE users tried various interaction technique combinations and expressed their opinion about the best choices to perform each task. The interaction techniques used in these studies were chosen from among the most commonly used and highly usable techniques described in the literature.

For the cooperative techniques, we based the separation of DOFs on the task to be performed, not to prove that those configurations are the best possible choices, but to demonstrate that the use of cooperative interaction techniques can be more efficient than two users working in parallel using single-user interaction techniques.

6.1. APPARATUS

In our studies, each user wore a tracked I-Glasses head-mounted display (HMD) and held a tracked pointer that allowed him to interact in the VE (Figure 13). To track the user’s hand we used a 6DOF Polhemus Fastrak tracker. Two separate computers were connected through their Ethernet interfaces in a peer-to-peer configuration at 10 Mbits/s, each one running the VE and having all the devices for a single user attached to it. In order to allow analysis of the users’ interaction their views of the VE were also displayed on two monitors that could be observed during the experiment by the evaluator.



Figure 13: Experimental apparatus

6.2. PROCEDURE AND METRICS

The experiments using the interaction techniques were performed according to a protocol that aimed at equally treating all the participating pairs. A group of 60 individuals (53 men and 7 women) participated in the

experiment organized in 30 pairs, not repeated. Ten pairs of users performed each experiment, and the tasks completion times were measured. The majority of the users were undergraduate and graduate students of Computer Science, who had good previous computer skills as well as experience with 3D graphics applications. The experiments did not have a minimum or maximum pre-established duration; however, the overall time for performing each experiment was between 50 and 65 minutes.

The protocol was divided into eight steps described below:

- (I) Applying the pre-test questionnaire: the users received a questionnaire asking about their age, activity, weekly frequency of computer use and previous knowledge of virtual environments.
- (II) Instructions about the experiment and the virtual reality equipment: the users received a sheet containing the description of the experiment and its objectives as well as the instructions. After reading the instructions, the equipment to be used during the experiment was presented to the users.
- (III) Presentation of the virtual environment: the users could observe (on the screens of both computers) the virtual environment they were going to use and the role of each device in that environment. The users were encouraged to manipulate their own glasses and pointers so that they could better feel the influence of those in the virtual environment.
- (IV) Training phase: the users wore their virtual reality equipment and they could freely interact in the virtual environment. At first, some basic instructions were provided in order to allow the preliminary exploration of the environment. Next, the individual interaction technique was presented to the users and they were asked to try it with objects within the virtual environment. Both users used the same interaction technique. During this training phase the users were introduced to the task they were to perform. From that point on they could practice for the individual execution of the task if they wished so. It is important to mention that individual execution does not mean that only one of the users could manipulate the object during a task. In fact, both could do it, but never simultaneously on the same object. At this time, the users were requested to develop a strategy for performing the task together. The users were encouraged to talk by using, whenever possible, the elements from the virtual environment itself, in order to demonstrate their ideas, strategies or intentions. The goal of this approach was to enhance even more the level of knowledge about the virtual environment and the users' feeling of presence in that environment. In this phase, there were frequently sentences like: "– You catch this object here and place it on the table. Then, I will manage to adjust it". Those sentences were invariably followed by the indication of the object through the use of the user's pointer. The virtual environment presented to the users in that step was the same that would later be used for the actual experiments.
- (V) Tests using the individual interaction technique: after the training session, the users were again "inserted" in the virtual environment and the task to be done was presented once again. The task performance was then timed. It is worth pointing out that the task execution was done in a collaborative way, but not simultaneously. The trial ended when a certain level of accuracy of object positioning and orientation was achieved. After completing this phase with the non-cooperative interaction technique, the users were asked to remove their glasses and to answer the first three parts of the evaluation questionnaire.
- (VI) Training for the collaborative technique: at this moment, the cooperative interaction technique was presented to the users. They could then test it for as long as they needed in order to feel comfortable with its use. The users were first requested to develop a strategy to perform the task together. In order to make the evaluation simpler, the users were asked to use the cooperative manipulation as much as possible.
- (VII) Tests using the cooperative metaphor: the users did their tasks in the cooperative way having their performance time measured once more. It is important to note that the configuration of the virtual environment (the initial position of the objects and the users) for this phase was the same as the one used for individual technique in the beginning of the manipulation phase using the individual technique. After finishing the task with the cooperative interaction technique the users were requested to take off their virtual reality glasses and to answer the rest of the evaluation questionnaire.
- (VIII) At the end of the experiment a quick informal interview was conducted with the users in order to find out if they had felt any kind of discomfort during the experiment, or if there was any additional comment they would like to make.

In the next three sections we provide a detailed description of each experiment.

6.3. EXPERIMENT WITH OBJECT DISPLACEMENT AND ORIENTATION

The first VE was designed with the purpose of evaluating the effect of cooperative techniques in the performance of users in tasks that required adjusting the position and orientation of objects. The VE simulates a

classroom in which two users (in opposite corners of the room) have to place computers on the desks. Figure 14 shows the view of one of the users.

The task was to place four computers on four desks in the middle of the room, in such a way that the computers had their screens facing the opposite side of the white board. This task involves both object movement and orientation.



Figure 14: Virtual Environment for experiment 1

For the individual execution of this task we chose the HOMER technique, because the task required two basic actions that are easy to perform with this technique: selection (and manipulation) of distant objects and rotation of objects around their local coordinate axes. After a pilot study, we decided to allow the user to slide the selected object along its pointing ray so that he could bring it closer or push it away from his hand – the indirect HOMER technique as described by Bowman [3].

The cooperative technique chosen for the simultaneous manipulation allowed one of the users to control the object's position and the other to control object's rotations. We have chosen this technique because we could clearly see the existence of two steps: one, when the object is moved from its initial position to the desk where it will be placed, and another, when the object is placed in its final position by means of small rotations. The control of the sliding function was disabled in the cooperative technique.

Each pair of users performed the task in a non-cooperative condition (each user used the individual technique, and the users divided the task between them), and a cooperative condition (the two users were allowed to use the cooperative manipulation technique). This is a strict test of the power of cooperative manipulation, because indirect HOMER has been shown to be appropriate and efficient for this type of task [3].

The experiment was conducted using ten pairs of users. Our pilot studies showed no effect of the order of the two conditions. This was because we let the users perform several training sessions before starting the tests, so that they achieved a high level of expertise.

Therefore, in the experiment we always asked the pairs to use the individual technique first, since the cooperative technique assumes that users already know the individual technique.

Table 3 shows the time taken to complete the task in the two conditions.

On average, the task time for the cooperative condition was two minutes and nine seconds less than in the non-cooperative condition. A t-test analysis indicated that this difference was highly significant ($p < 0.0001$).

Table 3: Comparison of time results obtained from experiment 1

Pair	Mean	STD Deviation
Without cooperative manipulation (min:sec)	06:45	2:16
With cooperative manipulation (min:sec)	04:36	1:46
Difference (min:sec)	02:09	
% Difference	32%	

6.4. EXPERIMENT WITH LARGE MOVEMENTS AND OBJECT FITTING

The second task designed to evaluate the use of cooperative manipulation consisted of placing objects inside some divisions of a shelf. Figure 15 shows what both users could see in the VE. With this experiment we aimed at evaluating the effect of collaborative techniques in situations involving the manipulation of distant objects, which are very common in virtual reality applications.

For performing the task with an individual technique the users used ray-casting with the sliding feature. At first we tested the HOMER technique, but we decided not to use it in this task because it did not present any significant advantage in the interaction process. In addition, HOMER is in fact more difficult, because it requires a complex control when the user needs to apply large movements to a selected object.

At the beginning of the experiment the objects were put next to user U1 and far away from the shelf. This user selected the desired object and put it next to the other user (U2), placing it in front of the shelf. At this point, user U2 was able to select and orient the object as wished, and could start moving it towards the shelf.

Because of the distance between user U2 and the shelf, depth perception was a problem when placing the objects. U1 could then give U2 advice to help him slide the object along the ray.

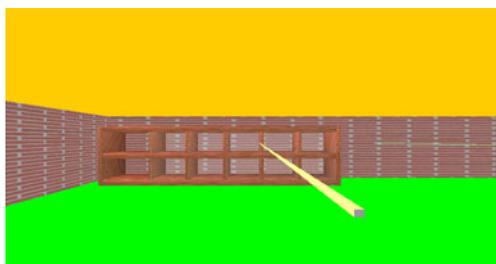
For performing the simultaneous manipulation for this task, we chose to configure the cooperative manipulation in such a way that the user placed in front of the shelf (U2) was able to control the objects' translation, leaving the sliding and rotation of the objects for U1, who was

close to their initial position. U1 did not control the movement of the object along his own ray, but along U2's ray. We have called this type of control remote sliding. This way the user in front of the shelves needed only to point into the cell where he wanted to place the object, while the other user controlled the insertion of the object into the shelf by sliding and rotating it.

The experiment was again conducted using ten pairs of users. Table 4 shows the resulting data from the individual and cooperative manipulation, considering the time for finishing this task. In this experiment, a t-test also indicated that the difference between both results was highly significant ($p < 0.001$).



User U1 view



User U2 view

Figure 15: Scenario for the shelf task

6.5. EXPERIMENT WITH MOVEMENT IN A CLUTTERED ENVIRONMENT

The third experiment fulfilled the goal of evaluating the use of cooperative tasks for manipulating objects in cluttered environments. We asked users to move a couch through an aisle full of columns and other obstacles. A user (U1) was at one end of the aisle (Figure 16) and the other one (U2) was outside, on the side of the aisle. For doing this task using an individual interaction technique, based on our pilot study, we again chose HOMER.

Table 4: Experiment 2 results comparison

Pair	Mean	STD Deviation
Without cooperative manipulation (min:sec)	07:12	02:43
With cooperative manipulation (min:sec)	03:56	01:20
Difference (min:sec)	03:16	
% Difference	42%	

For doing the task using the individual technique the users instinctively used a strategy in which U1 started the object's manipulation and, sliding it along its ray, placed it inside the aisle. Upon finding an obstacle, this user released the object, which was then selected by the partner (U2) who avoided the obstacle and released the object, giving control back to U1.

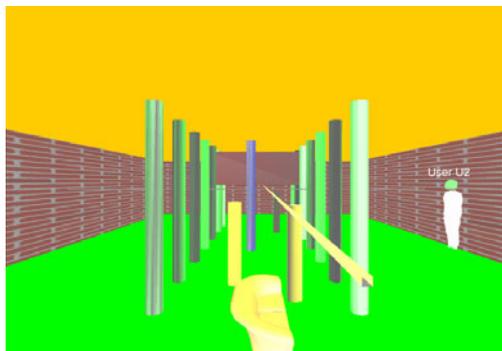


Figure 16: Scenario for experiment 3 (User U1 view)

The cooperative technique chosen for this task was configured to allow user U1 to control the object translations and user U2 the rotations and the remote slide.

The experiment was conducted using ten pairs of users. Table 5 shows the resulting data from the individual and cooperative manipulation, expressing the time for finishing this task. A t-test applied to these results also indicated that the 59-second difference between the means was highly significant ($p < 0.00001$).

Table 5: Experiment 3 results comparison

Pair	Mean	STD Deviation
Without cooperative manipulation (min:sec)	03:02	00:35
With cooperative manipulation (min:sec)	02:03	00:53
Difference (min:sec)	00:59	
% Difference	35%	

7. DISCUSSION

The experiments allowed us to evaluate both the architecture and different methods of combining individual techniques. It also allowed us to evaluate the basic premise for the work, that certain tasks are more easily performed using cooperative manipulation during collaboration when compared to non-cooperative manipulation methods (in which the users work in parallel).

Concerning the design of cooperative techniques, the experiments allowed us to verify different alternatives for separating the DOFs that each user can control.

Several configurations of positional DOFs were tested. The most common was the one in which a user could move the object in the horizontal plane while his partner controlled just the object's height. This technique was useful in cases where the users had to move objects among obstacles that were not all the same height. In such cases, while one user moved the object forward, backward and sideways, the other one simply lifted or lowered the object to avoid the obstacles. A similar configuration was important for controlling the movement of distant objects, particularly when one of the users could not clearly see the final position of the object. In the second experiment, for instance, the user in front of the shelf could not clearly see how much the object had to be moved forward or backward, in order to be correctly fit in one of the available spaces. The partner's cooperative manipulation allowed the correction in the final positioning phase.

The techniques that allowed one user to slide the object along his partner's ray also provided a greater control over small adjustments in the final position of the object. One of the users could point to where the object should be moved while the other controlled the sliding itself. This technique was particularly useful in those cases where the user who controlled the direction of the ray had a good view of the trajectory to be followed by the object, but was too distant from its final position. This technique is also applicable to other interaction techniques that use a ray as a pointer.

In order to be sure that our studies were not biased in favor of cooperative manipulation techniques due to poorly designed individual manipulation techniques, the latter ones were chosen among the most commonly used and highly usable techniques, and our users had a long training session.

The post-test questionnaire provided feedback regarding both the individual manipulation and cooperative techniques. The users reported that:

- The avatar facilitates the communication and the interaction because it allows knowing where the partner is, what he/she is doing and looking at (34 users, 56.66%);
- The use of the special designed awareness 3D icons was helpful for understanding the interaction capabilities of himself and the partner's (44 users, 73.333%);
- The use of HMDs provoked motion sickness (9 users, 15%) and eye strain or headache (26 users, 43.33%);
- The low HMD color resolution makes the precise object position a harder task (11 users, 18.33 %).

Finally, the experiments allow us to assert quite confidently that for many tasks in which the insertion of a collaborator (with non-cooperative manipulation) improves task execution, the use of simultaneous cooperative

manipulation provides an even greater benefit.

8. CONCLUSION AND FUTURE WORK

Our architecture and techniques are based on the separation of the control of degrees-of-freedom among the users, and provides several novel contributions to the field of collaborative VEs, including:

- Allowing the use of "magic" interaction techniques in cooperative manipulation processes,
- Allowing cooperative manipulation in immersive environments, and
- Cooperative manipulation without the need of force feedback devices.

In the beginning of this study we were concerned with the context change between the individual and collaborative activities (and how this would affect the users), a recurrent problem in collaborative environments. Separating the interactive techniques into distinct components made it possible to control the effect of the users' actions and to prevent the interference of the activity of one user into the other, regardless of the phase of the interaction.

Our architecture allows an easy configuration of the degrees of freedom controlled by each user, if it is based on techniques such as ray-casting, HOMER or Simple Virtual Hand. In these cases, the configuration is performed simply by changing a configuration file that defines the interaction technique and the DOFs controlled by each user during the cooperation. To include an individual technique that is totally different from the ones already implemented, we simply need to change the Interaction Module that interprets the movements of each user.

The cooperative techniques were implemented with minimum changes in the individual techniques. An important point to make is that the designation of the DOFs that each user will control is done a priori, in the configuration of the cooperative technique itself. The possible dynamic and immersive configuration of who controls each DOF is left for future work. The main problem in this case is how to build an immersive interface to perform such a configuration procedure.

Regarding the scalability of our approach, it is important to emphasize that although the tasks we designed for the experiments do not require higher DOFs, our approach can still be used with more complex tasks by grouping related degrees of freedom and assigning each group to a particular user.

Future work may involve studies of DOF coordination for multiple users, similar to previous research on two-handed interaction [33] [15], and experiments for comparing our approach to more recent

work in bimanual interaction [32]. As additional future work we also foresee performing experiments with physically separated users. By supporting awareness in different phases as well as avatars for partners we could measure the effect of not having spoken communication (although implementing CVEs with communication might become common soon).

Also, since the architecture does not limit the number of users that can participate in a collaborative interaction session, we plan to evaluate the usability of these techniques with more than two users.

Finally, it should be noticed that 3D applications are becoming common in many domains especially for training, but also for remote operation. Further studies on interaction within VEs and CVEs are necessary to help improving the quality of such human-computer interfaces.

ACKNOWLEDGMENTS

We thank the students from PUCRS which volunteered as subjects for our experiments and the graphical designer André Tomasi, for some of the illustrations we used in this paper. We also acknowledge the financial support from CAPES, CNPq and FAPERGS.

REFERENCES

- [1] C. Basdogan, C. Ho, M. A. Srinivasan, M. Slater. An Experimental Study on the Role of Touch in Shared Virtual Environments. *ACM Transactions on Computer-Human Interaction*, 7(4):443-460, 2000.
- [2] R. Bolt. Put-that-there: Voice and Gesture at the Graphic Interface. *Computer Graphics*, 14(3): 262-270, 1980
- [3] D. Bowman, L.F. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *Proceedings of 1997 Symposium on Interactive 3D Graphics (I3D)*. ACM Press, New York, p. 35-38, 1997.
- [4] D. Bowman, L.F. Hodges. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *The Journal of Visual Languages and Computing*, 10(1): 37-53, 1999.
- [5] D. Bowman, D. Johnson, L.F. Hodges. Testbed Evaluation of Virtual Environment Interaction Techniques. *Presence: Teleoperators and Virtual Environments*, 10(1):75-95, 2001.
- [6] D. Bowman, E. Kruijff, J. LaViola, I. Poupyrev. 3D User Interfaces: Theory and Practice. Addison-Wesley, Boston, 2004.
- [7] L.D. Cutler, B. Fröhlich, P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics (I3D)*. ACM Press, New York, p. 107-114, 1997.
- [8] T. Duval, A. Lecuyer, S. Thomas. SkeweR: a 3D Interaction Technique for 2-User Collaborative Manipulation of Objects in Virtual Environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI 2006)*, IEEE Computer Society Press, Los Alamitos, p. 69-72, 2006.
- [9] Y. Fabre, G. Pitel, D. Verna, D. Urbi et Orbi: Unusual Design and Implementation Choices for Distributed Virtual Environments. In *Proceedings of 6th International Conference on Virtual Systems and Multimedia (VSMM'2000)*, p. 714-724, 2000.
- [10] E. Frécon, M. Stenius. DIVE: A Scalable Network Architecture for Distributed Virtual Environments. *Distributed Systems Engineering Journal*, 5(3):91-100, 1998.
- [11] M. Goebel. Digital Storytelling – Creating Interactive Illusions with Avocado. In *Proceedings of International Conference on Artificial Reality and Telexistence (ICAT'99)*, Japan, p. 9-22, 1999.
- [12] C. Greenhalgh, S. Benford. MASSIVE: A Virtual Reality System for Tele-conferencing. *ACM Transactions on Computer Human Interfaces*, 2(3):239-261, 1995.
- [13] K. Hinckley, R. Pausch, D. Proffitt, N.F. Kassell. Two-handed virtual manipulation. *ACM Trans. Computer Human Interaction*, 5(3): 260-302, 1998.
- [14] G. Kessler, D. Bowman, L.F. Hodges. The Simple Virtual Environment Library: an Extensible Framework for Building VE Applications. *Presence: Teleoperators and Virtual Environments*, 9(2): 187-208, 2000.
- [15] A. Leganchuk, S. Zhai, W. Buxton. Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study. *ACM Transactions on Computer-Human Interaction*, 5(4):326-359, 1998.
- [16] M. Macedonia, M. Zyda, D. Pratt, P. Barham. Exploiting reality with multicast groups: A Network Architecture for Large-Scale Virtual Environments. In *Proceedings of IEEE Virtual Reality Annual International*

- Symposium (VRAIS'95)*, IEEE Computer Society Press, Los Alamitos, CA, p.15-19, 1995.
- [17] D. Margery, B. Arnaldi, N. Plouzeau. A General Framework for Cooperative Manipulation in Virtual Environments. In *Proceedings of the IEEE Virtual Environments' 99*, IEEE Computer Society Press, Los Alamitos, CA, p. 169-178, 1999.
- [18] M. Mine. Virtual environment interaction techniques. UNC Chapel Hill CS Dept. Technical Report TR95-018m, 1995.
- [19] M. Mine, F. Brooks, C. Sequin. Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction. In *Proceedings of the 1997 ACM Conference on Graphics (SIGGRAPH'97)*, p. 19-26, ACM Press, New York, 1997.
- [20] J. Mulder. Remote Object Translation Methods for Immersive Virtual Environments. In *Proceedings of Virtual Environments Conference & 4th Eurographics Workshop (EGVE'98)*, p. 80-89, 1998.
- [21] M. S. Pinho, D. Bowman, C.M.D.S. Freitas. Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques. In *Proceedings of ACM Virtual Reality Software and Technology (VRST 2002)*, ACM Press, New York, p. 171-178, 2002.
- [22] I. Poupyrev, M. Billinghurst, S. Weghorst, T. Ichikawa. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'96)*, p. 79-80, 1996.
- [23] R.A. Ruddle, J.C. Savage, D.M. Jones. Movement in Cluttered Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 10(5): 511-524, 2001.
- [24] R.A. Ruddle, J.C. Savage, D.M. Jones. Symmetric and Asymmetric Action Integration During Cooperative Object Manipulation in Virtual Environments. *ACM Transactions on Computer-Human Interaction*, 9(4):285-308, 2002.
- [25] R.A. Ruddle, J.C. Savage, D.M. Jones. Levels of control during a collaborative carrying task. *Presence: Teleoperators and Virtual Environments*, 12(2):140-155, 2003.
- [26] E.L. Sallnäs. Collaboration in Multimodal Virtual Worlds: Comparing Touch, Text, Voice and Video. In *Social Life of Avatars*. Schroeder, R.
- [27] S. Smith, D. Duke. Virtual Environments as Hybrid Systems. In *Proceedings of the 17th Eurographics Annual Conference (Eurographics'99)*, Eurographics/Blackwell, Oxford, p. 169-178, 1999.
- [28] R. Stoakley, M. Conway, R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of CHI'95*, ACM Press, New York, p. 265-272, 1995.
- [29] J. Viega, M. Conway, G. Williams, R. Pausch. 3D Magic Lenses. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST'96)*, ACM Press, New York, p. 51-58, 1996.
- [30] J. Wainer, C. Barsottini. Empirical research in CSCW - a review of the ACM/CSCW conferences from 1998 to 2004. *Journal of the Brazilian Computer Society*, 13(3):27-36, September 2008.
- [31] K. Watsen, M. Zyda. Bamboo - a Portable System for Dynamically Extensible, Real-time, Networked, Virtual Environments. In *Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS'98)*, p. 252-260, 1998.
- [32] H.P. Wyss, R. Bach, M. Bues. iSith - Intersection-based Spatial Interaction for Two Hands. In *Proceedings of IEEE Symposium on 3D User Interfaces (3DUI 2006)*, IEEE Computer Society Press, Los Alamitos, p. 69-72, 2006
- [33] S. Zhai, P. Milgram. Quantifying Coordination and Its Application to Evaluating 6 DOF Input Devices. In *Proceedings of CHI'98*, ACM Press, New York, p. 320-327, 1998.