University of California
Merced

Immersive Virtual Human Training Systems based on Direct Demonstration

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Carlo Camporesi

2015

The dissertation of Carlo Camporesi is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

Shawn Newsam

Alberto Cerpa

Marcelo Kallmann, Committee Chair

University of California, Merced

2015

To my parents Romano and Caterina ...

Acknowledgments

I would like to first express my sincere gratitude to my doctoral advisor and mentor, Professor Marcelo Kallmann. His patient guidance, teaching and insightful ideas inspired me to keep thinking out of the box and at the same time focusing my efforts on asking the correct questions. Through his supervision, expertise and guidance I have learned and developed skills that now reveal to be true assets for the development of my professional career. Evidently, I would not be here today without his support.

I would like to thank my committee members, Professor Alberto Cerpa and Professor Shawn Newsam, for the valuable inputs, suggestions, for the support they provided me on my research projects and for making important comments to the draft of this dissertation.

Next, I would like to thank my labmates, colleagues, and co-authors from UC Merced. Thank you Mentar Mahmudi, Yazhou (David) Huang, Robert Backman, Oktar Özgen, Paola di Giuseppantonio di Franco, Fabrizio Galeazzi, Anthony Popelar and Justin Matthews for your continued support but must importantly for your friendship. Some of the work included in this dissertation was only possible through our collaborations.

I would like to extend my gratitude to all the people closely or remotely involved in this project and a special thanks to all the participants and students involved during the experiments. In particular, I would like to express my gratitude to the Physical Medicine and Rehabilitation group of the University of California Davis Medical Center (UCDMC) especially Linda Johnson, Alina Nicorici, and Professor Jay J. Han for their continuous support during the development of the Virtual Physical Therapy projects. Additionally, I want to thank the UC Merced Graduate and Research Council, the Center for Information Technology Research In the Interest of Society (CITRIS), the National Science Foundation (NSF) and the HSRI San Joaquin Valley eHealth Network seed grant funded by AT&T for financially supporting my research.

I do not have words strong enough to say how grateful I am to all the many good people, all my friends, family and extended family that have supported me during this experience. Paola, Fabrizio (and little Edoardo that came along the way) we shared a roof for many years but the most important outcome is that you always make me feel at home. Thanks for being the two constants during this great experience abroad. To my “perfect virtual wife with hyper efficient organizational

skills" Estela Reinoso Maset and to my friends Justin Matthews, Collin Closek and Bjørn Larsen, thank you for the encouragement and the affect you always provided me when I most needed it. I am very grateful to Bryce Tahajian that took care of me during the most critical time preparing the thesis defense and the continuous support after this time. To all the great people I met and shared my life with during this journey, Marco Valesi, Alicia Ramos Jordan, Michael Romano, Mapi Asta, Nicola Lercari, Ana Becerril, Nicoló Dell'Unto, Maksym Vladymyrov, Nicole and Joshua Madfis, Gökce and Görkem Erinç, Ruthie Xochihua, Amanda Camelo, David Dunham, Bryan Kent, David Wright, Mark and Reyes Mullis and to all the many other friends I could not fit in this list, thank you for being there!

Finally, my most important thanks goes to my parents, Romano Camporesi and Caterina Sartini, I can not begin to thank you for all you have done over the years. I thank you for your endless support and unconditional love reaching to a level that no one can ever comprehend. Your sacrifices and encouragement have made everything I have done possible. This work is dedicated to you.

Vita

2015–Present	Software Engineer R&D, Avatire Inc., USA.
2008–2014	Graduate Student Researcher & Teaching Assistant, University of California Merced, USA.
Summer 2010	Senior Research Associate, School of Creative Media, City University of Hong Kong, HK-SAR.
2005–2008	Research Associate, ITABC - National Research Council, Italy.
2004–2005	Research Intern, Visual Information Technology Lab., CINECA, Bologna.
1998–2005	M.Sc., Computer Graphics, Alma Mater Studiorum University of Bologna, Italy.

Relevant Publications

Camporesi C., Kallmann M. - The Effects of Avatars, Stereo Vision and Display Size on Reaching and Motion Reproduction. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2015 (accepted)

Di Giuseppantonio Di Franco P., Camporesi C., Galeazzi F., Kallmann M. - 3D Printing and Immersive Visualization for Improved Perception and Interaction with Past Material Culture. In *Presence: Teleoperators and Virtual Environments*, 2015 (accepted)

Camporesi C., Kallmann M. - Computing Shortest Path Maps with GPU Shaders. In *Proceedings of the Seventh International Conference on Motion in Games (MIG)*, ACM, pages 97-102, 2014

Kallmann M., Camporesi C., Han J. - VR-Assisted Physical Rehabilitation: Adapting to the Needs of Therapists and Patients. In *Virtual Realities Dagstuhl Seminar 20013*, Springer, 2014

Camporesi C., Popelar A., Kallmann M., Han J. - Motion Parameterization and Adaptation Strategies for Virtual Therapists. In *Intelligent Virtual Agents (IVA) - Lecture Notes in Computer Science*, volume 8637, pages 99-108, 2014

Camporesi C., Kallmann M., Han J. - VR Solutions for Improving Physical Therapy. In *IEEE Virtual Reality (VR) - (poster paper)*, pages 77-78, 2013

Camporesi C., Kallmann M. - A Framework for Immersive VR and Full-Body Avatar Interaction. In *IEEE Virtual Reality (VR) - (poster paper)*, pages 79-80, 2013

Di Giuseppantonio Di Franco P., Camporesi C., Galeazzi F. - 3D Virtual Dig: a 3D Application for Teaching Fieldwork in Archaeology. In *Internet Archaeology: the premier e-journal for archaeology*, 2012

Kenderdine S., Camporesi C., Forte M. - Rhizome of Western Han Dynasty: an Omni-spatial theatre for archaeology. In *Computer Applications and Quantitative Methods in Archaeology (CAA)*, pages 141-158, 2011

Camporesi C., Huang Y., Kallmann M. - Interactive Motion Modeling and Parameterization by Direct Demonstration. In *Intelligent Virtual Agents (IVA) - Lecture Notes in Computer Science*, volume 6356, pages 77-90, 2010

Abstract of the Dissertation

Virtual humans have great potential to become as effective as human trainers in monitored, feedback-based, virtual environments for training and learning. Thanks to recent advances on motion capture devices and stereoscopic consumer displays, animated virtual characters can now realistically interact with users in a variety of applications. Interactive virtual humans are in particular suitable for training systems where human-oriented motion skills or human-conveyed information are key to the learning material.

This dissertation addresses the challenge of designing such training systems with the approach of motion modeling by direct demonstration and relying on immersive motion capture interfaces. In this way, experts in a training subject can directly demonstrate the needed motions in an intuitive way, until achieving the desired results.

An immersive full-scale motion modeling interface is proposed for enabling users to model generic parameterized actions by direct demonstration. The proposed interface is based on aligned clusters of example motions, which can be interactively built until coverage of the target environment. After demonstrating the needed motions, the virtual trainer is then able to synthesize motions that are similar to the provided examples and at the same time are parameterized to generic targets and constraints. Hence, autonomous virtual trainers can subsequently reproduce the motions in generic training environments with apprentice users learning the training subject. The presented systems were implemented in a new development middleware that is scalable to different hardware configurations, from low-cost solutions to multi-tile displays, and it is designed to support distributed collaborative immersive virtual environments with streamed full-body avatar interactions. An immersive full-scale motion modeling interface is proposed for enabling users to model generic parameterized actions by direct demonstration. The proposed interface is based on aligned clusters of example motions, which can be interactively built until coverage of the target environment. After demonstrating the needed motions, the virtual trainer is then able to synthesize motions that are similar to the provided examples and at the same time are parameterized to generic targets and constraints. Hence, autonomous virtual trainers can subsequently reproduce the motions in generic training environments with apprentice users learning the training subject. The presented systems were implemented in a new development middleware that is scalable to different hardware configurations, from low-cost solutions to multi-tile displays, and it is designed to support distributed collaborative immersive virtual environments with

streamed full-body avatar interactions.

Given the several possible configurations for the proposed systems, this dissertation also analyzes the effectiveness of virtual trainers with respect to different choices on display size, use of avatars, and use of user-perspective stereo vision. Several experiments were performed to collect motion data during task performance under different configurations. These experiments expose and quantify the benefits of using stereo vision and avatars in motion reproduction tasks and show that the use of avatars improves the quality of produced motions. In addition, the use of avatars produced increased attention to the avatar space, allowing users to better observe and address motion constraints and qualities with respect to virtual environments. However, direct interaction in user-perspective leads to tasks executed in less time and to targets more accurately reached. These and other trade-offs were quantied and performed in conditions not investigated before.

Finally, the proposed concepts were applied for the practical development of tools for delivering monitored upper-body physical therapy. New methods for exercise modeling, parameterization, and adaptation are presented in order to allow therapists to intuitively create, edit and re-use customized exercise programs that are responsive and adaptive to the needs of their patients. The proposed solutions were evaluated by therapists and demonstrate the suitability of the approach.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Virtual Humans are computer-generated interactive entities that should look and behave as much as possible like real people. Specifically, they should function autonomously, perceive their environment and react to events in the virtual (and real) world around them, always responding appropriately. They should interact in a fluid, natural way using the full collection of human verbal and nonverbal communication channels. They should model their own and others' beliefs, desires, and intentions and they should exhibit emotions. Ultimately, they should do all these things in a coherent, integrated fashion [337]. Virtual humans have great potential to become as effective as human trainers in monitored, feedback-based, virtual environments for training and learning. Interactive virtual humans are in particular suitable for training systems where human-oriented motion skills or human-conveyed information are key to the learning material.

The aim of this dissertation is to address the challenge of designing such training systems with the approach of motion modeling by direct demonstration and relying on immersive motion capture interfaces. Thus, experts in a training subject can directly demonstrate the needed motions in an intuitive way, until achieving the desired results. To achieve such a system I present in this dissertation an immersive full-scale motion modeling interface to model generic parameterized actions by direct demonstration where, subsequently, autonomous virtual trainers can reproduce the motions in generic training environments with apprentice users learning the training subject. Given the several possible configurations for the proposed systems, the effectiveness of virtual trainers with respect to different choices on use of avatars, and use of 3d vision was analyzed. Finally, the proposed concepts were applied to the application of delivering monitored upper-body physical therapy. The proposed solutions were evaluated by therapists and demonstrate the suitability of the approach.

1.1 Motivation

The task of constructing Virtual Humans is complex and requires extensive efforts. Many researchers have focused on this goal over the past decades. Building a virtual human requires integrated expertise from different fields and concerted efforts from researchers working at cutting edge computer graphics simulation technologies and from psychologists, linguists, cognitive scientists, and as well expert artists and animators. Such variety of disciplines is needed in order to create realistic virtual humans exhibiting autonomy, natural communication, and sophisticated reasoning and behavior.

While the ultimate goal of building the *perfect* virtual human remains a challenging topic, significant progress has been made. Virtual humans have been studied and applied in different contexts, such as: dialogue and nonverbal behavior [342, 56], animation and gestures [240, 22, 95], action and dialogue [66, 287, 111]. A common point across these works is to achieve virtual humans that can interact effectively with real people but still, many trade-offs related to the employment of virtual humans are still open questions. Understanding how users perceive and interact with virtual characters in several application design conditions is not clear and many questions and settings need to be addressed in order to develop novel and effective applications.

Animated characters, avatars and virtual humans often emerge as key elements in approaches focusing on replicating human forms of communication. In particular, virtual characters play a significant role in motion-oriented training and rehabilitation applications. Virtual Reality training systems started to appear as soon as the first Virtual Environment prototypes came to life. Virtual Environments, Immersive Systems and Augmented Reality have been extensively applied to train users in a variety of tasks. These new training environments have been reported to be successfully perceived and to help both trainees and trainers in their roles. A wide variety of training applications have been developed; for example, applications to simulate hazardous military situations [276, 168], training operators to use specific machinery [308, 139, 373] or to train surgeons in medical procedures [201, 309].

Virtual humans have been applied in training scenarios to convey different types of information more efficiently, proving to be important in fulfilling the distance between a real human trainer and a sterile learning environment based only on presentation of visual information. This gap is prominent in tasks where human-oriented skills, such as leadership, negotiation, cultural awareness, and human-conveyed actions (such as body-language, motions and gestures) must be delivered during the

learning process. These skills are based on what is called tacit knowledge [329], that is, knowledge that is not easily explicated or taught in a classroom setting but instead is best learned through experience. Currently, these training experiences are usually delivered through various human-to-human role-playing exercises and through imitation. Virtual trainers can effectively replace the human role in many situations leading to several benefits. Human-based role playing is costly in terms of personnel requirements and is often done at training centers that may be far away from the student's location. In contrast, virtual exercises can be delivered on any computer device, making them available to a student whenever and wherever they are needed, without the need to add additional personnel resources.

Following this approach several training systems have been proposed. For example Traum et al. [353] proposed a prototype with the goal of exposing junior army officers to delicate situations that might occur during the course of a mission but that typically are not covered in a standard training manual or course (e.g. helping civilians during a targeted mission). Beyond rehearsal, another human-oriented skill explored is the possibility of creating virtual humans that could negotiate [354]. Similarly, behavioral and negotiation training has been proposed in the context of a work environment where the virtual human occupies a position of authority [290].

The systems described above are complex and they require to integrate a broad range of technologies, including: speech recognition; natural language understanding and generation; dialogue management; task and domain reasoning; emotion modeling; speech synthesis; and gesture generation. All of these modules rely on a strong human component that involves a large amount of programming effort and time. In this process, it is crucial to model the virtual tutor's behaviors with the ability to convey information to the trainee in the same manner as a real tutor would convey it, retaining the original detail and motions' properties of the expert tutor. As humans we rely on our innate and profound capability of learning through imitation. The imitation learning process, sometimes similar to Programming by Demonstrations (PdB) or Learning from Demonstration (LfD) [75, 242], has been addressed frequently in Robotics and the same approach can be extended to embrace virtual training scenarios to enforce behaviors and specifics. In Robotics, the main challenge consists on the problem of learning a mapping between world state and actions (*policy*). This policy enables a robot to select an action based upon its current world state. Within Imitation learning, a policy is learned from examples, or demonstrations, provided by a teacher. Demonstrations can be provided in an indirect manner (where there is no embodiment mapping between the robot and the teacher) or in direct form (where an embodiment mapping between the robot and the teacher exists). Generally speaking, examples are considered as sequences of state-action pairs

that are recorded during the teacher's demonstration of the desired robot behavior. LfD algorithms utilize this dataset of examples to derive a policy that reproduces the demonstrated behavior.

As previously mentioned, considering the situation where a virtual avatar needs to convey specific actions and gestures to a trainee, what is important is to offer to the users a situation where, in the modeling phase, experts in the training subject are able to model actions and gestures, without the need of having previous experience with the system. The stored example motions can be re-used by the virtual human during a training phase with non-expert users and they can be used to naturally demonstrate motions, and to monitor and analyze user performances.

In light of these needs the Direct Demonstration approach appears to be the most suitable option since it relies only on the subject's expertise and dexterity. Another important benefit of using pre-recorded motions is to maintain the original motions properties and qualities. Several techniques can be adopted in this phase to give to the virtual tutor enough liberty to reproduce these motions with respect, for example, to arbitrary target locations in new environments and according to different constraints. Therefore, the original motions have to be deformed and adapted, always under the supervision and acceptance of the application expert, in order to enable virtual humans to automatically deliver a correct and effective learning experiences. Considering this latter case the problem of adapting and conveying information to users through gestures can be redesigned from a generic decision learning task (where an action-state policy is no longer necessary) to a motion parameterization problem that combines realism, flexibility, precise control and ability to adapt to different environments.

In practice, achieving effective implementations of such virtual human training systems requires key design choices to be made and several key factors needs to be taken into account, for instance: the modeling and training phases have to be quickly interchangeable to allow a fast content modeling and testing; the adoption of a full scale immersive reality is necessary to grant one-to-one feasible interaction; or, while in some scenarios it may be useful for the user to see his or her own motions replicated in an avatar, in some other scenarios avatars may in fact distract the user from paying attention to the task at hand. The most appropriate configuration may also depend on hardware choices. For example, large full-scale screens, small desktop screens, and displays with stereo vision influence user performances in avatar-based scenarios in different ways. Achieving effective implementations of such systems therefore requires a deeper understanding of the tradeoffs involved among the many possible configurations.

Virtual humans are also key in collaborative environments with remote participants, where a remote instructor may control a local avatar delivering training material to users. Human instructors can also pre-design training plans to be later performed autonomously by a virtual character. Such case is usually a wide adopted approach in traditional physical therapy where, after a preliminary step of diagnostic and quantitative measurement a patient is guided by a trained therapist to perform specific therapeutic exercises. The tasks performed are designed according to a recovery plan, which implies repetitions of exercises and constant progress evaluation both qualitatively and quantitatively. The traditional physical therapy process is usually intensive, time consuming and dependent on the expertise of the therapist. It also implies the collaboration of the patient who is usually asked to perform the therapy program multiple times at home with no supervision [19, 250]. Patients often perceive the tasks as repetitive and non-engaging, consequently reducing their level of involvement [127, 177]. This fact is related to a number of aspects: lack of customization on how to execute exercises, communication and interaction practices that are unsuitable to a particular patient, no clear perception of improvement, lack of coaching and monitoring while at home, etc. Addressing these many aspects is important to improve training sessions or therapy outcomes, and in particular to reduce the risk of injuries due to wrongly executed exercises. Even in this case, motion demonstration methodologies are suitable to be successfully adopted, allowing therapists to intuitively create, edit and re-use customized exercises that are responsive adaptively to the physical limitations and recovery rates of individual patients.

1.2 Objective and Contributions

The overall objective of this dissertation is to create an effective pipeline for the development of virtual human training systems with the ability to demonstrate, instruct and deliver training information in immersive virtual environments.

The techniques and models proposed by this pipeline are developed considering specific objectives such as: intuitively collect motion capture data from expert users on a specific field for high-fidelity synthesis (direct demonstration approach); precise parametrization for specified constraints; direct interaction through full-scale immersive visualization; employ real-time algorithms for fast responsiveness and real-time user interaction and distant collaboration; real observations of users to delineate application's key design choices when avatars and immersive visualization is adopted; and, adoption of these techniques on a real training and monitoring scenario such as rehabilitation and physical therapy.

The work presented in this dissertation introduces new approaches, techniques and evaluations which contribute to the achievement of these goals. The specific contributions are briefly summarized as follow:

- I propose an immersive full-scale motion modeling platform for enabling users to model generic parameterized actions by direct demonstration. The interface proposed allows experts to interactively build training motion dataset in real-time by demonstration. The demonstrated motions can be performed by users directly with respect to specific virtual environments. After demonstrating the needed motions, the virtual trainer is then able to synthesize new motions for teaching and training purposes for novel users. The motions are parameterized to generic targets, constraints and they can be reproduced in environments different from where they where recorded.

- I performed several experiments, collecting motion data and user evaluations, to analyze the effectiveness of virtual trainers with respect to different choices on display size, use of avatars, and use of 3D vision. The results show evidence that the use of user-perspective stereo vision with direct interaction is the optimal choice in terms of task accuracy and completion time, when precision tasks are involved (6 times more accurate and 2 times faster) and also improved replication of spatial relationships, both in terms of synchronization and compliance with the reference motion, even when the task was transferred to the avatar's space (by a factor of 3). Coupling avatar use with stereo vision resulted in users paying more attention to the motions within the virtual environment, improving the realism and correctness of the motions. In addition, avatar use showed to well improve motion reproduction in the cases where stereo vision is not present. This factor suggests that in application setups where stereo vision is not practical to be used (such as in homes or clinics), the use of avatars will definitively improve the user understanding of motions displayed by virtual tutors. These and other results were quantified and were performed in conditions not investigated before in previous work.

- I applied the previously described concepts and guidelines to a concrete scenario, developing a set of applications and tools to deliver monitored upper-body physical therapy. New methods for exercise modeling, parameterization, and adaptation are presented in order to allow therapists to intuitively create, edit and re-use customized exercise programs that are responsive to the needs of their patients. In addition, several other factors can be considered to adjust a system to the user's preferences: from the display of messages, instructions

and videos, to the appearance of the virtual character demonstrating the exercises, etc. The system also includes remote networked solutions for allowing therapists and patients to share motion performances in real-time. The transmitted data is lightweight and remote collaboration can well scale to several patients at the same time. The proposed solutions were evaluated by therapists and demonstrate the suitability of the approach.

In addition to the main contributions of this dissertation, the presented systems and applications were implemented in a new development middleware designed to facilitate the creation and deployment of Collaborative Immersive Virtual Reality applications with full-body avatar interactions. The framework is scalable to different hardware configurations and it defines high-level programming abstractions to hide, simplify and automate several aspects of the creation of distributed virtual reality applications. The presented framework demonstrate new interaction paradigms for immersive training systems and the system has been tested and compared with other analogous solutions demonstrating similar performances without the use of customized hardware. The framework is also based only on open source libraries or with source code available.

The results of this research will be useful directly in fields such as virtual reality, computer graphics, computer animation and robotics. In addition, the use of virtual humans employing motions from real-motion capture information will impact areas where the general understanding and perception of human motion is involved such as psychology, cognitive science and HCI. Finally, as it will be presented in several chapters of this dissertation, this approach will simplify and improve the delivery and presentation of human motion practices and can therefore have an impact on education, training and the medical area.

1.3 Overview of Chapters

The remainder of this dissertation is organized as follows:

Chapter 2 presents a broad literature review on the fields related to this dissertation: virtual reality infrastructures; character animation; real-time motion reconstruction; motion perception; and virtual reality applications, with particular emphases on Rehabilitation and Physical Therapy.

Chapter 3 describes the development middleware to create Collaborative Immersive Virtual Reality applications with full-body avatar interactions. The Chapter

describes, at first, the overall system structure, modules and dependencies involved highlighting key properties of the system such as: scalability to generic hardware configurations; integrated with high-end graphics tools; high-level programming abstractions; inclusion of advanced tools for full-body character animation, real-time full-body tracking, and motion transfer to virtual humans and so on. Following a brief overview of some of the developed projects and application, performance testing results are proposed.

Chapter 4 introduces the methodology and a framework, based on the direct demonstration and parameterization of motions, to empower nonskilled animators with the ability to program generic actions and gestures for virtual trainers using a full scale immersive visualization system. The chapter presents the overall concept explaining all the implementation details and methodologies involved.

Chapter 5 investigates the influence of different system configurations on user performances during task execution in immersive virtual environments employing the motion modeling concept. The investigated configurations varied with respect to display size, the use of stereo vision, and the use of avatars, in which case the task is transferred to the avatar space (with the avatar mimicking the motions of the user). The presented systematic study also investigates the use of user-perspective stereo vision both in direct interaction mode and in 3rd-person mode with the avatar being driven by the user to perform a given task. The Chapter finally highlights benefits and gives guidelines for the design of applications using direct demonstration and interaction with 3d vision.

Chapter 6 presents a complete system for the delivery of upper-body physical therapy. The system described uses motion demonstration and parameterization methodologies to allow therapists to intuitively create, edit and re-use customized exercises that are responsive to the needs of their patients. The illustrated system also present remote networked solutions for allowing therapists and patients to share motion performances in real-time. The chapter also presents all the techniques and tools developed for the monitoring, assessment and visualization of a number of therapy parameters during or after execution of exercises.

Chapter 7 summarizes the dissertation findings together with an analysis of the promising directions for future research.

Given its close relation to the central topic of this dissertation (and in general to other important areas such as virtual reality, human perception, and character animation) but not directly related to the main topic of motion modeling by demonstration, the following work is left in the Appendix:

Appendix A investigates the advantages of 3D immersive stereoscopic system and 3D prints for interacting with past material culture over traditional observation of the material without manipulation. Results to these experiments suggest that traditional museum displays limit the experience with past material culture, and reveal how our sample of participants favor engagement, tactile or 3D virtual experience with artifacts over a visual non-manipulative experience with authentic objects.

CHAPTER 2

Literature Review

This chapter reviews the state of the art of the research related to this dissertation. The presented review provides denitions of basic concepts which are important for the understanding of the virtual reality field and its history. Virtual Reality systems and related research are then examined with particular emphases on the technologies adopted, system design differences and related applications. Subsequently, the chapter describes the efforts achieved in the character animation field considering the problem of solving real-time motion reconstruction from users and direct mapping on virtual avatar. Following evaluation studies on perception and usability of virtual reality systems are then discussed. Finally, a comparison and digression about VR applications, with particular emphases on Rehabilitation and Physical Therapy, are presented.

2.1 Virtual Reality

The term virtual reality has found its origin during 1960s but a more formal definition was proposed by Howard Rheingold [286]. Rheinghold's view defines virtual reality (VR) as an experience in which a person is "surrounded by a three-dimensional computer-generated representation, and is able to move around in the virtual world and see it from different angles, to reach into it, grab it, and reshape it."

At the present, even though many progress have been achieved toward the development of VR technologies that can be easily adopted in home environments (such as: Oculus Rift; Leap and Kinect sensors; Sixsense Stem; etc.), technology is still not ready to make computer generated worlds as believable as reality in the same way science fiction or scientific VR forecasts were depicting in the past decades. Virtual Reality uses computers to create 3D environments in which one can navigate and interact with the specific goal of simulating reality and the ability to navigate and interact in computer generated scenes with the possibility of creating reproductions of real objects or environments for training, entertainment or design purposes [114].

In this review and in the rest of this work we adopt the concept and the classification of VR provided by Milgram et al. [225]. In their manuscript they proposed the idea of a Reality-Virtuality Continuum, a way to understand and classify different types of display technologies. The concept is illustrated in Figure 2.1.

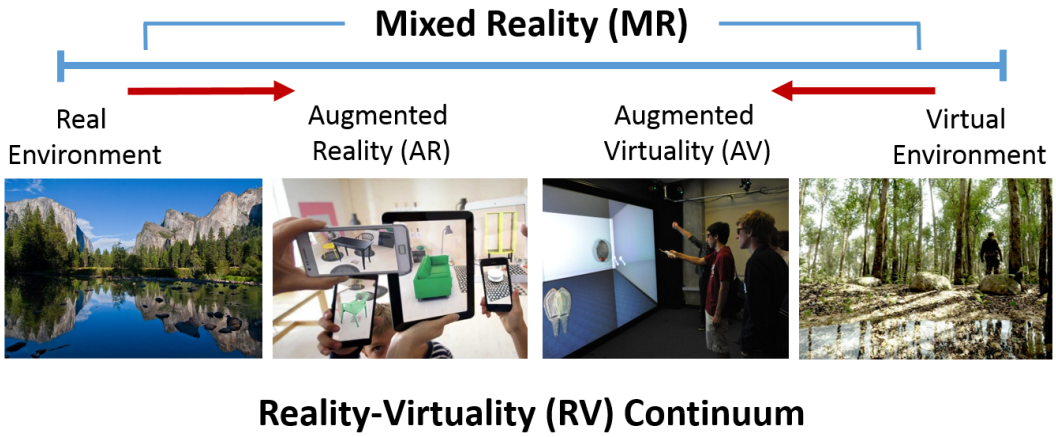


Figure 2.1: Simplifed representation of Reality-Virtuality (RV) Continuum.

The case at the left of the continuum defines any environment consisting solely of real objects, and includes whatever might be observed when viewing a real-world scene either directly in person, or through some kind of a window, or via some sort of a (video) display. In the Real Continuum no artificial computer generated stimuli are provided to the user. The case at the right defines environments consisting solely of virtual objects, examples of which would include conventional computer graphic simulations, either monitor-based or immersive. In this latter case everything the user can perceive is artificial. Within this framework it is straight forward to define a generic Mixed Reality (MR) environment as one in which real world and virtual world objects are presented together within a single display. More specifically, in augmented reality, most of the images are real while in augmented virtuality, most of the imagery is computer-generated.

Another source of confusion regarding the concept of VR is whether or how it is different from 3D games. Broadly speaking, games can be considered a subcategory of VR systems that are focused on the task of entertaining players but keeping the interface and interaction mechanism simple and well known. However, in the current years, we are starting to witness the interest and adoption from the major game and entertainment companies of sophisticated, more advanced and affordable

technologies for visualization [258], interaction [224, 186] and user sensing [224, 332]. The products delivered are becoming closer to what a virtual reality application requires.

The aspects of reality that have been most prominently explored are the visual and the tracking ones but simulating reality implies different other important components. For example other important components of our perceptual experience are sounds and tactile feedback. Developing VR systems involves different disciplines that address each of the human senses: computer graphics (sight), 3D sound synthesis (hearing), and haptics (touch). Smell and taste play important roles in our daily life. Several projects to reproduce smell in VR have been proposed but taste has been less explored due to the complexity of the required technology.

In this line of thinking, three main factors have been identified as the main contributors that enhance the VR experience from a physical and psychological points of view: presence, immersion and multi-modal interaction.

Presence is defined as the degree to which participants subjectively feel and process cues translating them in a coherent environment in which we can perform some activities and interact. According to Slater and Wilbur "Presence is a state of consciousness, the (psychological) sense of being in the virtual environment" [322]. In order to promote presence in a synthetic experience the following aspects must be considered: sensory fidelity and richness; degrees of interactivity; and other psychological cues [259, 313]. A well-crafted virtual world could change our emotional state and make us feel differently according to the type of experience designed.

Immersion is defined by the physical possibility to perceive the virtual environment as in real life through stereoscopy and spatial sound. Immersion is directly related to the physical configuration of the VR application. VR systems can be classified as fully immersive (at least visual and auditory cues are fulfilled by the system enforcing user isolation from the real world), semi-immersive (visual cues are fulfilled almost as close as a real experience without isolation), or non-immersive (any other virtual experience). The challenge for computer graphics researchers is to make the virtual world look, sound and respond to user interaction in real-time [366]. A comprehensive description of immersive visualization technologies and hardware can be found in [245].

Multi-modal interaction comprehends any type of natural and usable mechanism (and related user interfaces) to promote user machine interaction. According to [160], from intuition and from the evolutionary viewpoint, an interaction scheme is based on: the three-dimensional space; involvement of the whole body; and one that takes advantage of the multimodality of the human sensory system.

2.1.1 Virtual Reality history

During the 1960s, as previously mentioned, the first real system to simulate the real world was developed. The system was developed by the cinematographer Morton Heilig in 1962 and its name was "Sensorama". It was designed to simulate vehicles and it allowed users to sit in front of a screen where they could choose from different vehicles. The system did not provide any interactivity but did include 3D photographic slides, stereo sound and even a wind generator.

In 1965, Ivan Sutherland during his lecture entitled The Ultimate Display [335] in which he described how one day the computer would provide a window into virtual worlds. He suggested that a screen should be considered as window upon a virtual world. The real world is complex and it contains many stimuli which involve many senses. Subsequently, he built a head-mounted display (HMD) that presented to the user left and right views of a computer-generated 3D scene. After these initial experiments Virtual Reality research started to grow finding as main contributors Computer Scientists as well as Artists. For example, a few years later, the contribution of Myron Kruger with a gesture based interactive art project "Videoplace" [175] started the idea of interactive virtual reality or "artificial reality".

In order to finally find the first system that can be considered a true Virtual Reality system we have to move to the 1980s where innovations were supported by greater accessibility to 16-bit CPUs. The world of VR was at one of its peaks right around 1984 where the Ames Research Center at NASA developed the first pilot training system for manned space missions VIEW [368]. The VIEW system consisted of a wide-angle stereoscopic display unit, glove-like devices for multiple degree-of-freedom tactile input, speech-recognition technology, gesture-tracking devices, 3D audio, speech synthesis, computer graphics, and video image generation equipment.

These years mark the maturity of many aspects of computer technology and, to many degrees, the structural foundation for modern VR/AR technologies. The first initiatives to commercialize VR products started in the early 1980s. VPL Research was one of the first companies focused on developing VR hardware and software. VPL developed the "DataGlove" integrated with head-mounted displays. Subsequently, other companies arose specializing their business in different aspects from interaction, force-feedback, tracking and pure visualization. As lead companies we find names like: Polhemus; Ascension; Virtual Technologies; and Immersion Corp.

Even though the 1980s are considered the peak of VR research and commercialization the technology involved and the computational power of machine was not ready to fulfill the needs of complex VR applications. During the 1990s the design of

the first CAVE system [73] changed the paradigm of Virtual Reality offering a setup less cumbersome than current HMD, easier to interact, providing a large field of view and depth cues. The CAVE system is based on back-projecting images on walls surrounding the user. This setup allowed for the first time multi-user immersion and interaction in a shared virtual space.

Around the 2010s, thanks to the technological advancements and the involvement of major companies (such as Microsoft, Facebook, Intel, Apple, Sony, etc.), it seems we have finally arrived at affordable, high-quality consumer hardware for Virtual Reality. The game and entertainment market is finally at the stage of adopting more sophisticated methods of interaction and visualization for their products such has Oculus RIFT [258], Sixense [319], Virtuix [362] and Microsoft Kinect [224].

2.1.2 General purpose Immersive VR architectures

Since the early 1980s numerous systems have been proposed that aimed to develop engines to create VR application and advanced systems. More specifically, immersive VR systems (iVR) like CAVEs and multi-tile walls (or Powerwalls) started to appear in the 90s and have defined what can be called cluster-based virtual reality.

In the early stage of development, due to lack of hardware power and high task specialization, the most common approach chosen consisted of adapting each solution to Domain-Specific cases (these packages are focusing to specific uses and requirements or are monolithic packages that offer little flexibility to developers [34, 197]) or to create Special-Purpose Architectures to solve a task. For example, high-performance real-time rendering systems have relied on an integrated proprietary system architecture, such as the SGI graphics supercomputers [310] or specialized solutions like PixelFlow [230], MPC Compositor [235], Sepia [203] or Lightning 2 [330].

In the following years, this high specialization of both software and hardware lead researchers toward the development of high level generic infrastructures with the goal of simplifying the workload of programmer and decoupling the application implementation details from the hardware where the application is running on. A large number of VR development software ranging from low-level programming toolkits to content creation systems aimed at non-programmers arouse after the 1990s.

Following, a list of the most representative engines for the development of VR system is provided, both commercial and open source. Each system provides different mechanism abstraction in regards to: display type support (3D Displays, Projectors, Head-Mounted Displays, etc.), rendering techniques (single-pipe, multi-pipe

Figure 2.2: Examples of Virtual Reality setups. On the left: CAVE system with four retroprojected walls using MiddleVR and Unity [141, 355]. On the Right: Carina Nebula displayed in a multiframe LCD powerwall system using CGLX [82].

and cluster-based) and input abstraction.

Single-pipe and Multi-pipe rendering techniques rely on a single machine/workstation connected to one or more rendering/projection devices. This approach is becoming more common thanks of the development of more powerful computers and multi-display graphics cards [255, 20]. On the other hands when really high resolution surfaces are needed or when the computational power of a single workstation is not enough to generate virtual simulations cluster-based rendering is still the main solution.

A number of algorithms and systems for parallel rendering have been developed in the past. Cluster-based systems can be subdivided in two main categories:

Client-Server (CS) systems have the data and the application state maintained by the server and subsequently each update is streamed to the clients on demand (data streaming, distributed rendering calls or rendered buffer streaming). For example, specific algorithms have been developed for cluster-based rendering and compositing such as [4, 27, 385].

Master-Slave (MS) systems maintain the data mirrored or shared between parallelized application instances and the application status is maintained synchronized among all instances. Some general concepts applicable to cluster parallel rendering have been presented in [232, 233] (sort-first architecture), [302, 301] (load balancing), [300] (data replication), or [58] (scalability).

2.1.2.1 Master-Slave architectures

Of particular note is the CAVELib [73, 72, 57] the first general purpose VR engine developed that introduced the concept of CAVE. Initial versions of its immersive projection environment ran on a pair of networked SGI computers. Consequently, the CAVELib includes support for PCs and commodity clusters. CAVELib hides display setup by running individual render processes for each screen. Applications provide callback functions that insert user specific OpenGL code and in a cluster, application instances are generated in all nodes and synchronizes them by sharing input signals. CAVELib does not provide any high level abstraction and constructs to the user making the generation of clustered based applications quite complex.

VR Juggler [35] is a a VR toolkit with a level of abstraction similar to CAVELib which shields the application developer from the underlying hardware architecture, devices, and operating system. Display abstraction is based on the notion of projection surfaces. Two extensions provide clustering support, NetJuggler [9] and ClusterJuggler [260]. Both extensions are communicating with the VR Juggler instance and they are based on the replication of application and data on each cluster node. These extensions basically take care of synchronization issues but fail to provide a flexible configuration mechanism that efficiently supports scalable rendering. VR Juggler contains no high-level geometry library, high level graphics programming constructs or 3D GUI components. Although VR Juggler supports clusters, the user still needs to develop applications considering low level server/client implementations as also noted by [327].

Other two extensions of VRJuggler has been proposed during the past few years trying to address the main limitations of VRJuggler. The FlowVR platform [10] was developed based on experience in using VR Juggler in a clustered environment, and emphasizes a data-flow model for distributed real-time interactive computation with high modularity. VRJugg LUA [263] was designed a high-level virtual reality application framework based on LUA scripting language, VR Juggler and OpenSceneGraph. This approach allowed the generation of VR applications for distributed rendering with the enhancement of a high level Graphics Library. The system design is still cumbersome and the abstractions mechanisms are not fully developed.

Another popular VR library is Syzygy [305]. Syzygy is a programming toolkit aimed at commodity clusters with a limited multipipe abstraction. The library either offers a high level scene graph architecture, or cluster-based rendering very similar to ClusterJuggler. In comparison to VRJuggler, the library offers a high level abstraction level considering networked applications but it does not offer higher-level interactions. Simulators (especially on the desktop) are still severely limited.

HECTOR [369] supports the idea of rapidly prototyping applications using a Python scripting interface. The framework consists of a micro kernel and a communication module similar to VRJuggler. Many central components like the event propagation system are directly implemented in Python, which makes the system easily extendible. This framework proposes an interesting concept, but it seems that this project is no longer active. Similarly, AVANGO/NG [176] applies a generic field and field container programming interface to a scenegraph based on OpenSceneGraph (originally using SGI Performer), with Python scripting support.

3DVia Virtools used to be a widely used commercial development kit for 3D applications that supports many common hardware components and comes with a lot of built-in functionality. The VR Library/Publisher module supports client-server distribution where the parameters of a scene are synchronized with the server, but it does not support a peer-to-peer communication of stand-alone VR/AR applications. The project is now discontinued.

Worth noting, Unity Technologies released a plug-in [141] to run the Unity3D software [355] on any VR platform, providing at the same time its vast game designer network and Asset Store to VR research community. Along with Eon Reality [281] and Worldviz [379] solutions (Eon Icube and Vizard respectively), commercial solutions are nowadays available to support researchers in their VR based investigations, on the understanding that they are disposed to invest financially in such, typical installations having yearly maintenance fees most often calculated on a per screen basis. This limits development flexibility and mutliplatform/multi-configuration utilization. The relatively closed nature of these solutions also limits their use in research oriented applications.

2.1.2.2 Client-Server architectures

One of the first fully-fledged engines, providing high level abstractions and cluster based rendering, for virtual reality is Chromium [137] (the successor of WireGL [136]). Chrominum provides a powerful and transparent abstraction of the OpenGL API, which allows a flexible configuration of display resources. Its main limitation with respect to scalable rendering is that it is focused on streaming OpenGL commands through a network of nodes, often initiated from a single source (commands and data).

The OpenGL Multipipe SDK (MPK) [33] implements an effective parallel rendering API for a shared memory multi-CPU/GPU system. MPK handles multipipe rendering by a lean abstraction layer via a conceptual callback mechanism, and that

it runs different application tasks in parallel. However, MPK is not designed nor meant for rendering nodes separated by a network. MPK focuses on providing a parallel rendering framework for a single applications that are running in parallel on multiple rendering channels.

Parallel rendering frameworks such as Garuda [246] and Aura [358] show the potential of the parallel approach but are restricted to a scene graph-type application.

A system that addressed the limitations of the previously mentioned systems is Equalizer [87]. The Equalizer framework does not impose these restrictions and provides enhanced load balancing strategies targeted mainly toward virtual reality-type systems. However, to leverage these features, in depth knowledge about the Equalizer system configurations and considerable code changes or application specific adaptations are required. With the same implementation philosophy of Equalizer CGLX [82] was developed. CGLX aims to provide an easy-to-use parallel rendering framework for distributed cooperative visualization systems without imposing any restrictions on OpenGL-type applications. CGLX gives users full control over the display configuration. Hardware-accelerated rendering is natively supported on different operating systems.

A totally different approach is proposed by the Scalable Adaptive Graphics Environment (SAGE) architecture [148]. SAGE operates on the assumption that any type of application will send a pixel stream to the SAGE server, which in turn manages the tiles and distributes the incoming pixels to the correct portion of a tiled wall. SAGE takes exclusive control of the distributed frame buffer. Thus, to display a high-resolution visual, another application needs to be running on the same cluster, rendering its content in an off-screen buffer, which then can be read back and mapped to a SAGE client. Since read-back operations are expensive, the achievable performance of this approach is limited.

2.1.3 Distributed Virtual Environments

A distributed virtual environment (DVE), as defined in [190], is a software system that allows users in a network to interact with each other by sharing a common view of their states. As users are geographically distributed over large networks like the Internet and the number of users increases, scalability is a key aspect to consider for real-time interaction.

The approaches for improving scalability of DVE can be classified into the following categories [114]: communication architecture, interest management, concurrency states control, data replication, and load distribution.

Communication architecture the communication architecture can be characterized as follows: client-server (messages are sent to a server and then the server distributes them to all clients); peer-to-peer (users directly exchange messages and maintain synchronization); or peer-to-server (consistency management is done by a server and communication among users is performed using multicast).

Interest management Interest-management strategies [30] exploit the fact that users do not need to receive all update messages related to the whole world. Instead, they receive only messages in which they are interested. Two possible scenarios are feasible: world regions (zones) or adaptive radius localization.

Concurrency states control Shared information in DVEs is often replicated at each user's site to provide acceptable interactive performance. Replication enables users to locally access and update data. On the other hand data has to be maintained synchronized, which eventually leads to inconsistent views among users. Approaches to concurrency control have been broadly categorized into pessimistic, optimistic and prediction schemes (most frequently used for spatial positioning changes and updates).

Data replication Data is replicated by local changes or notification of remote changes. Three scenarios are possible: preloaded local data; monolithic shared data download; and on-line data streaming. The main challenge in partial data replication is the minimization of scene artifacts due to unavailable data (or partially loaded data). Two schemes are usually used together: prioritized transfer of objects and caching (e.g. LODs and CLODs mechanisms [200]) and prefetching techniques.

Load distribution Partitioning a virtual world into multiple regions and distributing the responsibilities for managing the regions across multiple servers can significantly reduce the workloads of individual servers. To avoid degradation of interactive performance, due to imbalance among servers, dynamic load distribution schemes have been introduced where overloaded servers transfer their excessive workloads to less-loaded ones. There are three main approaches: local (the workload is shared among neighboring servers), global (a main coordinator balances the workloads in the whole grid), and adaptative dynamic load distribution (overloaded server balances its workload with a set of servers, beyond its neighboring servers, according to their workload status).

2.2 Kinematic-Based Character Animation

This section reviews the state of the art in skeletal character animation with specific emphasis on kinematic-based approaches useful for real-time motion reconstruction, blending and parameterization.

Kinematic-Based Character Animation can be subdivided in two main categories: data-driven and algorithmic approaches. Data-driven are based on set of prerecorded motions stored in labeled or unlabeled databases. The motions reproduce details and the style of human movement in a very accurate manner. There are many ways for capturing and recording skeletal-based motions including mechanical, magnetic, optical, and inertial devices. Each motion capturing (mocap) technology has its own strengths and weaknesses with regard to accuracy, expressiveness, and operating expenses. See [213, 229] for an exhaustive overview. Readapting these motions is a difficult task since they are tight to the original skeleton morphology that they were recorded with respect to a specific environment. Moreover, data-driven approaches can have the limitation of handling and maintaining specific constraints if the constraints are not represented in the span of the initial database (for example reconstructing a reaching motion of the hand in a place where it is not covered by any spatial interpolation of the initial database). Algorithmic methodologies can use different techniques, from optimization methods to more complex physics based solutions. These solutions are usually enough to simulate robot movements producing accurate and enforcing constrains but they are not sufficient to maintain the feasibility and the naturalness of a human motion. Complex implementations are usually very computational expensive and they might not be applicable in real-time.

Generic methods for modeling and generating human motions have been extensively studied in bio-mechanics, robotics and computer graphics. In the literature we can find a wide variety of methods to generate character motions based on a set of controllers. These methods are usually stable and quite suitable for robotics purposes but they might lack on certain human-like characteristics in the generated motions. For example the work proposed by [41] generates human postural control with a Prioritized Inverse Kinematics framework. Their method introduces a new type of analytic constraint called Flexion-EXTension constraint to solve and stabilize partial position control of limb-like articulated structures. Ho et al. [123] proposed a method to efficiently synthesize character motions that involve close contacts such as wearing a T-shirt, passing the arms through the strings of a knapsack, or piggy-back carrying an injured person. Their solution introduces the concept of topology coordinates, in which the topological relationships of the segments are embedded into specific attributes.

Another example is the whole-body analytical Inverse Kinematic method proposed by [154]. In this approach full-body animations of reaching tasks are produced integrating collision avoidance and customizable body control. The method relies on search algorithms for achieving specific postures avoiding joint limits and collisions. Also the problem of generically solving walking sequences and gait has also been addressed. A common method uses path following and sagittal elevation angle controls coupled with inverse motion synthesis based on barycentric interpolation [334].

As the technology for motion capture becomes more available, the attention has been focused on how to re-use the captured data to generate new motion sequences to precisely impose modifications and constraints. In this domain a well-known and active area of research is motion retargeting of motion capture data to different skeletal characters with or without similar topologies. [124] uses the new concept of interaction mesh, a structure to represent implicit spatial relationships between body parts and surrounding objects, to minimize local mesh deformations and interpenetrations within animation frames. Hecker et al. [118] introduces a novel way to record animations in a morphology independent form. At runtime the system uses an IK solver to animate characters with different skeleton morphologies. On the same line of research Yamane et al. [384] propose an alternative method for animating non-humanoid characters that leverages motion data from a human subject performing in the style of the target character. The method consists of a statistical mapping function learned from a small set of corresponding key poses, and a physics-based optimization process to improve the physical realism.

2.2.1 Motion Graphs

In this section Motion Blending and Reconstruction techniques are addressed with respect to techniques data-driven and algorithmic approaches since they are more suitable for real-time purposes.

One of the first approaches developed consists in synthesizing new motions by reordering the motion clips. This type of technique is well know as motion graph [171, 170]. Motion graph is based on connecting similar frames in a database of motion capture examples. Once a motion graph is available, graph search is performed in order to extract motions with desired properties. New motion sequences, such as certain styles of locomotion along arbitrary paths, are generated piece-by-piece by simply "walking" along the graph nodes and selecting those with minimal errors. Since the motion pieces selected are not modified, the final motion retains the details of the original motion data but the synthesized motions are restricted to those in the

motion capture database.

Arikan and Forsyth [14] build a hierarchy of graphs and use a randomized search to satisfy user constraints. Arikan et al. [15] use dynamic programming to search for motions satisfying user annotations. Lee et al. [191] construct a cluster forest of similar frames in order to improve the motion search efficiency. All these methods require quadratic construction time for comparing the similarity between all the frames in the database.

Other extensions of motion graph are focused on improving the search versatility. Heck et al. [117] introduced the new data structure of parametric motion graph, which is capable of dynamically generating the transitions in real-time. Interpolation methods have also been used. The method proposed by [298] is based on the interpolation of two time-scaled paths and involve methods that solve a linear system of constraints. Cheng et al. [284] proposed a method using optimization techniques and probability based transitions in the graph to achieve interactive natural transitions between motion clips. Although these methods increase the solution space, they come with the expense of further distorting synthesized motions or they often increase the involved computation time and complexity. Methods employing an Inverse Branch Kinematics to deform the 2D graph branch with Cyclic Coordinate Descent (CCD) solver, so that the branch end reaches the goal with much better precision [212], have also been explored.

In addition to the generation of motion graphs that can adapt to different tasks and databases, work on generating better motion graphs (small, fast, and with good connectivity as well as smooth transitions) have been proposed, for example, using iterative sub-graph algorithm has proposed by [390, 391].

2.2.2 Motion Parameterization and Reconstruction

Early studies in motion parameterization found their root in the signal processing analysis domain. Motions are considered propagation of signals over time and different techniques, such as, motion filtering, multitarget interpolation, Fourier analysis etc. have been used to extrapolate data, interpolate different motions or to extract and to alter motion styles [356, 370, 48].

Following the signal processing approach a different method to analyze motion data was proposed by Rose et al. [295]. This approach combines Radial Basis Function (RBF) with polynomial terms to generate new synthesis with examples characterized by emotional expressiveness or control behaviors. A draw back of this approach is found when the desired input gets further from the original data resulting

to degraded motions.

An extension of this work was proposed in [296]. The presented method applies an efficient inverse-kinematics methodology based on the RBFs interpolation of example motions and positions. RBFs can smoothly interpolate given motion examples and the types and shapes of the basis functions are optimized in order to better satisfy the constraints.

Spatial properties such as feet sliding or hand placements are well addressed by the geostatistical interpolation method [234], which computes optimal interpolation kernels in accordance with statistical observations correlating the control parameters and the motion samples. Another approach for improving the maintenance of spatial constraints is to adaptively add pseudoexamples [170] in order to better cover the continuous space of the constraint. This random sampling approach however requires significant computation and storage in order to meet constraints accurately and is not suited for handling several constraints. Moreover, the error between the end-effector and specified target still exists. Methods to correct this artifact have been proposed by [71] through learning feature and IK solver correction.

The Scaled Gaussian Process Latent Variable Model (SGPLVM)[110] provides a specific framework targeting the IK problem which optimizes interpolation kernels specifically for generating plausible poses from constrained curves such as positional trajectories of endeffectors. Similarly, [195] presents a technique that interactively animates characters performing user-specified tasks with constraints on a low-dimensional latent space. These methods are stable and produce reliable motions, however, the higher computational cost in the training process imposes limitations on applications that involve on-line appending or refining of the dataset.

The problem of parameterizing motions is linked closely to the problem of reconstructing character motions from a low resolution input signals (for example low resolution signal matched to a high resolution database or reconstruction of motion from a reduced marker set motion tracking device) or from a input signal that is different from a motion database (reconstruction of human motions from video camera, accelerometers or IR depth information). In addition, the (real-time) control of virtual characters using mocap data, also known as computer puppetry [314], is another key challenge in the field of computer animation. Besides the use of high-dimensional optical systems, various controller-based systems have been described that allow for generating and reconstructing visually appealing motion sequences on the basis of low-dimensional sensor input.

Once a motion database is parameterized, for example with the methods previously described, human motions can be reconstructed through search methods. Chai

and Hodgins [59] proposed a method to reconstruct user motion in real-time from a reduced and noisy subset of tracked markers. Their method relies on a graph of nearest neighbors for fast search of the motion examples that are close to the current control signals and use it to build a local linear model of the motion for interpolation. A similar approach was chosen by Zheng et al. [392] using hidden Markov models (Windowed Viterbi algorithm) to achieve the similar results. Another similar approach was proposed by Tautges et al. [343]. Instead of using the same set of input signals (in Chai's case marker based motion capture data) with a reduced number of inputs the authors are reconstructing full-body human walking using accelerometers.

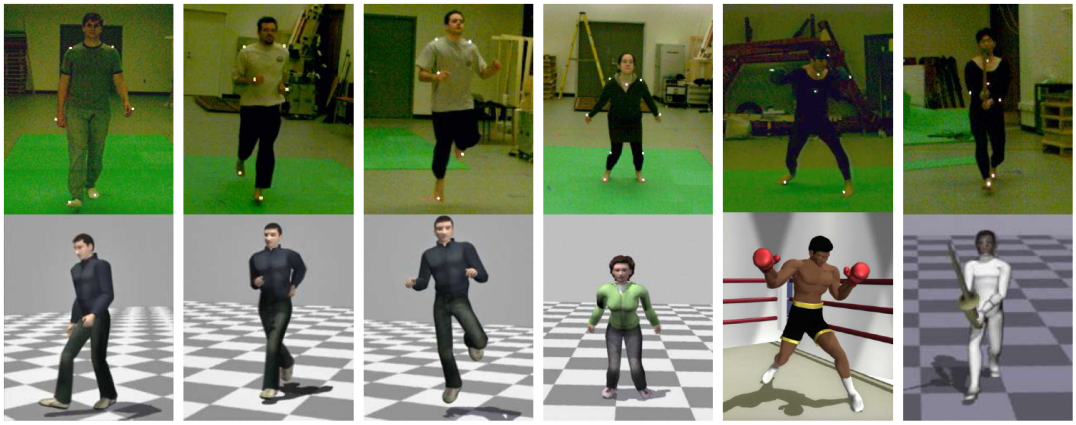


Figure 2.3: Character's motion reconstruction from a reduced marker set proposed by Chai et al. [59].

In Shiratori et at. [315], inertial-based control data is used to specify a small number of free parameters in physically-based character animation. When high-dimension data has to be generated using only low-dimensional control data data-driven approaches especially show promising results. This work has been subsequently extended using body-mounted cameras to reconstruct the motion of a subject [316].

Human pose estimation from camera based sensors has generated a vast literature (surveyed in [318, 272]). This problem is mostly addressed as a machine learning problem where, considering a vast number of training set data, new data is parameterized and the closest data from the data base is then retrieved considering some accuracy distance. A large number of papers based on this approach improved the regression models, dealt with occlusion and ambiguity, and re-incorporated temporal information [239, 357]. Such models, however, classify the whole-body pose in one

monolithic function and they require enormous amounts of training data.

One approach to reduce the demand for training data is to divide the body into parts and attempt to combine the per-part estimates to produce a single pose estimate. The first method that allowed the reconstruction system to work in real-time, fit to several body types and not requiring any calibration technique was proposed by [317]. Shotton et al.'s method predicts human pose from a single depth image, without depending on information from preceding frames. By designing an intermediate representation in terms of body parts, the difficult pose estimation problem is transformed into a simpler per-pixel classification problem, for which efficient machine learning techniques exist.

Le et al. [185] propose a method based on Laplacian mesh editing [326], which has proven to be well suited for animating close interactions. Here, interaction meshes [124] are defined with respect to their temporal correlations in the original animation. Motion optimization is achieved by solving two minimization problems. The first penalizes deformations that result from Laplacian coordinate manipulation and the second preserves the length of motion segments over time.

Ho et al. [122] proposed a method based on two-person motion capture data with a two stage process. First, the postures of the afterwards active interaction partner, i.e the human, and the virtual agent are organized in a kd-tree. This leads to a tree where each leaf stores pairs of poses that have been obtained in the initial recording. Then, for live human-agent interactions the tree is queried for postures that are similar to the current user pose. The character's pose is morphed to match spatial constraints by solving a space-time optimization problem with interaction meshes [124]. In doing so a virtual character can react to an ongoing human interaction in real-time.

On the same line of research Vogt et al. [363] proposed a new scheme for efficient identification of motions in the prerecorded animation data that are similar to the live interaction. A global low-dimensional posture space serves to select the most similar interaction example, while local, more detail-rich posture spaces are used to identify poses closely matching the human motion. Using the interaction mesh of the selected motion example, an animation can then be synthesized that takes into account both spatial and temporal similarities between the prerecorded and live interactions.

2.2.3 Learning from Demonstration

Learning from Demonstration (LfD), also referred to as Imitation Learning or Programming by Demonstration (PdB), has become a central topic in the robotics field in the past years. LfD spans across general research areas such as human-robot interaction, machine learning, machine vision and motor control.

LfD find its roots from the Programming by Example concept [75]. Programming by Example is defined as a way of programming a software system in its own *user interface*. The user of the system writes a program by giving an example of what the program should do. The system records the sequence of actions, and can perform it again. Programming by example allows a user to create programs without doing conventional programming. In Robotics, Programming by Demonstrations is considered a powerful mechanism for reducing the complexity of search spaces for learning. When observing either good or bad examples, one can reduce the search for a possible solution, by either starting the search from the observed good solution (local optima), or conversely, by eliminating from the search space what is known as a bad solution. Imitation learning is, thus, a powerful tool for enhancing and accelerating the learning process.

A wide variety of systems employ the Programming by Demonstration concept. Within LfD, a policy is learned from examples, or demonstrations, provided by a teacher. Examples are defined as sequences of state action pairs that are recorded during the teacher's demonstration of the desired robot behavior. LfD algorithms utilize this dataset of examples to derive a *policy* that reproduces the demonstrated behavior. This approach to obtaining a policy is in contrast to other techniques in which a policy is learned from experience, for example building a policy based on data acquired through exploration, as in Reinforcement Learning [26].

There are certain aspects of LfD which are common among all applications to date. One is the fact that a teacher demonstrates execution of a desired behavior. The second is that the learner is provided with a set of these demonstrations, and from them derives a policy able to reproduce the demonstrated behavior. The demonstration phase (recording) can be performed in two manners:

Presentation or indirect demonstration where there is no embodiment mapping between the robot and the teacher, because demonstration is performed on the actual robot learner. In this case two scenarios are possible: teleoperation and shadowing. Teleoperation is a demonstration technique in which the teacher operates the robot learner platform and the robot's sensors record the execution. The recording operation can be done via different media and used in a variety of applications, for

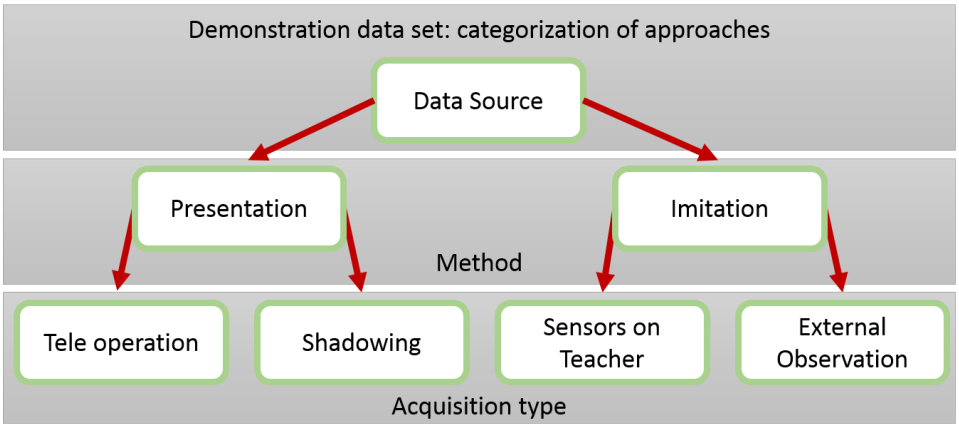


Figure 2.4: Learning from Demonstration: categorization of approaches to building the demonstration dataset as proposed Argall et al. [13].

example: via joystick to flying a robotic helicopter [243], object grasping [338, 24], robotic arm assembly tasks [65] or through kinesthetic teaching where robot's joints are passive and moved through desired motions [43, 12]. The shadowing technique consists in the robot platform mimicking the teacher's demonstrated motions while recording from its own sensors [76, 241].

Imitation or direct demonstration where there exists an embodiment mapping between the robot and the teacher, because demonstration is performed on a platform which is not the robot learner [303]. Also in this case two techniques can be applied: through sensor on a teacher and by external observations. In the first technique human teachers commonly use their own bodies to perform example executions by wearing sensors able to record the person's state and actions or by using sensors able to infer the user's pose. This is especially true when working with humanoid or anthropomorphic robots, since the body of the robot resembles that of a human. Example of application of technique has on robot walking [237, 304] or simulating human which maps to a simulated robot and then to a real robot arm [52] or full-body [108]. The external observation technique relies on data recorded by sensors located externally to the executing platform, meaning that a record mapping exists [270, 63]. The data recorded under this technique is less precise and less reliable. The method, however, is more general and is not limited by the overhead of specialized sensors and settings.

Given a dataset of state action examples several methods for deriving a policy using this data have been developed. In the Robotics domain the most common

technique applied relies on: learning an approximation to the state-action mapping (mapping function), or learning a model of the world dynamics and deriving a policy from this information (system model). Alternately, a sequence of actions can be produced by a planner after learning a model of action *pre* and *post* conditions (plans). Across all of these learning techniques, minimal parameter tuning and fast learning times requiring few training examples are desirable. Since the policy learning step is out of the scope of this dissertation the reader is referred to the extensive reviews on the field presented by the following manuscripts [36, 13, 51].

2.3 Avatar Perception in Virtual Reality

Humans, being highly social creatures, rely heavily on the ability to perceive what others are doing and to infer from gestures and expressions what others may be intending to do. The use of avatars (i.e., digital representations of human users) to represent individuals within multi-user immersive virtual environments in real-time has increased considerably in the past decade, underscoring the need to better understand how users experience them visually and cognitively.

Humans communicate and interact in complex ways and through different communication channels. An exhaustive overview of the perceptual, motoric, affective, and neural concomitants involved in human communication processes are documented by Blake [38]. Psychologists have showed that people are remarkably skilled in recognizing the features of a person through his or her motions; for example, in identifying gender [172], emotions [21], or the identity of a known person from just a synthesized silhouette of his or her motions [74].

Many aspects about character animation and avatars have been studied in the context of virtual reality applications: different rendering styles, behavioral or animation qualities of characters, user's perception of objects, embodied cognition, depth perception and visual cues and so on.

For example, the introduction of visual artifacts or the use of an unnatural animation has been demonstrated to lead to human negative reactions [45]. Reitsma et al. [282] presented the results of a study on user sensitivity to errors in animated human motions, and proposed a metric for measuring errors in ballistic human motions. McDonnel et al. [216] analyzed the impact that rendering styles of animated characters have on users, defining possible outcomes per scenario. Other types of studies have been performed targeting aspects like how the appearance of characters affect the perception of their actions [60] or the perception of bodily emotions [217].

The use of virtual avatars in immersive virtual environments has also been extensively explored and applied in many fields with investigation on aspects related to behavior, appearance, interaction, user's embodiment and ownership [39, 328, 265]. Recent work has examined how people come to "inhabit" or embody their avatars, which are virtual representations of themselves [184]. For example, Slater and colleagues [321] demonstrated that male participants experienced a so-called "body transfer illusion" even when their avatars were female. Several studies have demonstrated the persuasive advantage of using virtual selves over generic or unfamiliar virtual others in the realms of advertising [2], health [91], and financial decisions [121].

One recent study, comparing the persuasive power of IVE with character embodiment against traditional media messages, found that traditional IVEs induced/increased social behavioral intentions more effectively that traditional media [3].

The study proposed by Phillips et al. [267] discovered that users who are given a first person avatar in an Non-photorealistically rendered (NPR) immersive virtual environments replica judge distances more accurately than users who experience the NPR replica room without an embodiment. Moreover, the latter are not as accurate as users whose distance judgments are made in a photorealistically rendered virtual replica room.

In traditional Immersive Virtual Reality it has been found that humans are not reliable on distance and depth estimation in virtual reality [16]. A commonly accepted theory behind depth perception in virtual reality is based on the Depth Cue Theory [98], which is based on the concept of visual cues.

There are a number of cues that the human visual system uses that result in a perception of depth. Some of these are present even in two dimensional images, for example: perspective projection, size of known objects, details, occlusion, lighting and shadows etc. There are other cues that are not present in 2D images, they are: binocular disparity, accommodation (muscle tension needed to change the focal length of the eye lens in order to focus at a particular depth), and convergence (muscle tension required to rotate each eye so that it is facing the focal point). While binocular disparity is considered the dominant depth cue in most people, if the other cues are presented incorrectly they can have a strong detrimental effect.

According to Gilliam [99], depth perception results from the integration of visual cues categorized as: pictorial, oculomotor, and stereo depth cues. A more recent extension of this concept was proposed by Swan et al. [336], and states that not only the individual visual cues but also the ways these cues are processed by the human visual system allows us to create a three-dimensional perceptual world.

For the purpose of assessing depth perception cues (and the ability to estimate user's depth perception), two types of measurements are considered reliable: the allocentric (distances between objects and a reference point) and the egocentric (distances between objects in relation to the subject) distance [204]. The problem of distance perception in virtual environments is often related to two possible scenarios: peripersonal (space that can be reached by our hands) and the extrapersonal space. Research has showed that depth acuity is usually high in peripersonal space but low otherwise [275].

In this thesis work, instead of investigating the impact of different rendering styles, behavioral or animation qualities on users, the main focus is on investigating how the use of avatars mimicking the user's motions influence the successful execution of motor tasks and motion reproduction considering different virtual reality configurations. The key motivation of this research is driven by the importance of these tasks in several training and rehabilitation applications.

2.4 Rehabilitation and Physical Therapy

Healthcare and health science are one of the biggest adopters of virtual reality which encompasses many type of applications. For example there are many applications in surgery simulation, phobia treatment, post traumatic stress disorder treatment, robotic surgery, nursing, rehabilitation and so on. It is also used to train the next generation of doctors, paramedics and other medical personnel.

The coupling of VR-based new technologies with healthcare have been proven to bring many advantages in different aspects of the healthcare pipeline that both benefits patients and practitioners. Professionals can learn new skills as well as refreshing existing ones in a safe environment, without causing any danger to the patients. Distance collaboration and supervision can be adopted during complex procedures. Visualization technologies allows therapists to better analyze data and scans in an interactive and progressive manner enhancing and speeding up the diagnosis step. Patients can be advised by Virtual Physician during daily routines and they can be sensed and monitored while unsupervised from a real clinician. The development of technologies that can be applied to rehabilitation offers tremendous promise for enhancing functional capacity by eliminating or minimizing the functional limitations imposed by disability.

Several areas of Health Science have been addressed by Virtual Reality. The following sections address only the aspects and works related to VR applied to rehabilitation and physical therapy since a comprehensive overview of all the efforts achieved

in VR health science is out of the scope of this dissertation. For a broader overview on Health Science related sub-fields the reader is referred to [129, 158, 6, 114, 194, 273].

2.4.1 Physiotherapy

VR-based rehabilitation involves multiple disciplines, including physical therapy, computer science, psychology and communication. VR systems for rehabilitation integrate haptics and modern sensor technology [120]. Different level of VR-based rehabilitation systems have been developed during the years targeting different types of target audiences. More advanced systems used stereoscopic displays, force-feedback devices (haptic interfaces), and modern sensing techniques to capture accurate data for further analysis. These system have been designed as prototypes to be employed in clinics and high-tech facilities [113]. Other prototypes have been designed for a broader scale adoption using simple and inexpensive devices [288, 144]. Much attention has been paid to interactive digital games (i.e. Serious Games) that can provide numerous assets for rehabilitation beyond what is currently available with traditional methods [367].

Strategies incorporating the use of various technologies for the people with motor disabilities have been developed for occupational therapy across numerous settings. In particular, virtual reality and motion-based games have been used for rehabilitation recently. Industrial motion sensors [17, 312] and, in particular, entertainment oriented ones [23] such as Nintendo Wii Remote [5, 311] show research evidence that they are useful as physical rehabilitation tools. Furthermore, motion-based games that combine motion sensor technology and fun with video games can motivate people to engage in exercises that games design purposefully [5]. An extensive survey about human motion tracking for rehabilitation is provided by [393].

A drawback with motion sensors is that people have to fasten them on limbs, hold them in the hands or even wear them on the body to detect motions and possibly generate force feedback. Wearing sensors can cause inconvenience and discomfort. Computer Vision based technologies started to arise in the past few years. With the introduction of depth sensors, such as Microsoft Kinect or similar, a marker-less yet stable tracking technique has been improved the horizon of Virtual Rehabilitation.

Chang et al. [62] introduced the possibility of rehabilitating two young adults with motor impairments using a Kinect-based system in a public school setting. The two participants significantly increased their motivation for physical rehabilitation, thus improving exercise performance during the intervention phases. Lange et al. [180, 183] developed a Kinect-based rehabilitation game to encourage the players to

reach out of their static balance base of support. The approach proposed by Roy et al. [297] targets the physical therapy of patients with motor disabilities stemming from cerebral palsy, spinal cord injury, post stroke and hereditary muscle ailments amongst others.

Several research projects address the motor rehabilitation needs of post-stroke patients demonstrating the positive effects of serious games with virtual reality technologies on motor-skill improvement for functional deficits such as: reaching [128], hand function [221], and walking [388]. Jung et al. [152] proposed a project to develop a virtual reality enhanced stroke rehabilitation system (VRSRS). The system introduces a human factors design in developing a VRSRS for improving the functional recovery rate of stroke patients' upper extremities. Another example, using haptic interfaces was proposed by [77]. Their interface provides 6-DOF resistive forces on the patient's foot, in response to Virtual Realitybased exercises. This kind of physiotherapy has demonstrated to be effective and patients improved on clinical measures of strength and endurance after very few sessions.

The combination of haptic interfaces, Web technologies for data transmission, 3D graphics, and other multi-modal stimuli can be used to develop tele-rehabilitation systems. The basic concept is to provide the patient with an interface that will allow us to perform the physiotherapy designed and monitored by a doctor or therapist in a remote location.

An example of such a system is the tele-rehabilitation application presented in [113]. The system is designed to interface a virtual environment, controlled and monitored by a therapist in a remote location, with a patient using an haptic device. The patient performs exercises defined by the therapist. A virtual window displaying live video allows the patient to keep visual contact with people in the remote location. In addition, the therapist can design new exercises by means of drawing lines on a PDA; such lines will be transformed into 3D shapes with haptic feedback. The patient will be able to touch and interact with the 3D shapes, relearning arm coordination skills. Another example of the effectiveness of tele-rehabilitation systems is proposed by Piqueras et al. [268] addressing the problem of monitoring and providing a system for distant rehabilitation of patients after post knee arthroplasty surgery. On virtual tele-rehabilitation and doctor-patient distanct interaction Kurillo et al. [177] developed a tele-immersion prototype for tele-rehabilitation using real-time stereo vision and virtual environments. Stereo reconstruction is used to capture a user's 3D avatar in real-time and project it into a shared virtual environment, enabling a patient and therapist to interact remotely. Captured data can also be used to analyze the movement and provide feedback to the patient.

CHAPTER 3

Immersive Motion Modeling Infrastructure

Collaborative and Immersive Virtual Reality (VR) is becoming increasingly popular in many areas and several software frameworks have been proposed to support its development. Current solutions are however designed around specific hardware, designed to solve specific problems, and/or designed as a commercial solution. In addition, a gap can be often observed between the available capabilities and what is offered by state of the art graphics engines in terms of qualitative rendering, performance, or high-level programming tools available to the developer [151].

We present in this chapter a new development middleware to create Collaborative Immersive Virtual Reality applications addressing most of these limitations and based on one of the most successful open source game-oriented rendering engines OGRE [345]. The framework combines a modern open source game rendering engine with the flexibility to develop and run executable from simple desktop computers to multi-tiled cave systems in a completely transparent and simple fashion (using commodity or more expensive hardware systems).

The system implementation design defines high-level programming abstractions to hide, simplify and automate most of the system and device related calls and distributed programming architecture requirements, such as: object state sharing, internal synchronization, object replication, etc. In addition, the system also includes a communication layer designed to enable collaborative interaction between application instances over the Web. Finally, the system is scalable to generic hardware configurations, is integrated with high-end graphics tools, and includes advanced tools for full-body character animation, real-time full-body tracking, and motion transfer to virtual humans. Such capabilities enable novel full-body interaction metaphors for collaborative VR.

Since this solution is completely designed on top of cross-platform Open-Source code (or source code available), the core maintenance process becomes possible and the system can be adapted or extended to specific or particular implementations needs.

In the following paragraphs we present a related work overview and the framework

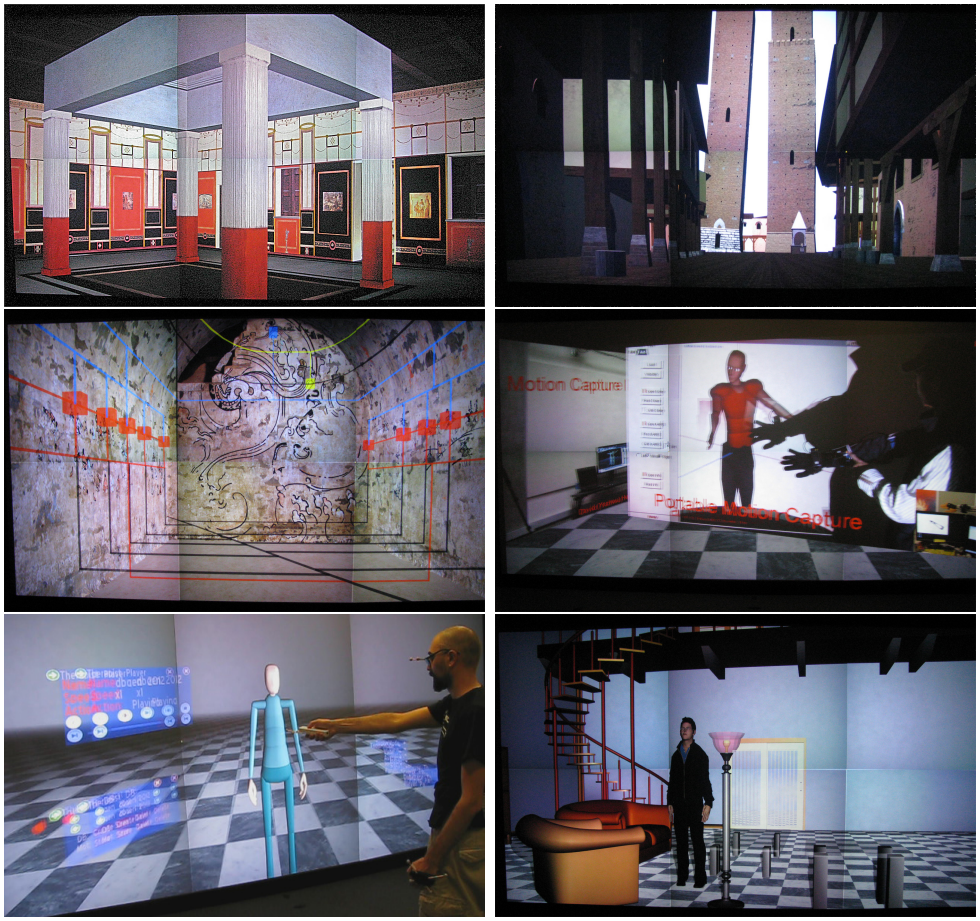


Figure 3.1: The framework running in a low-cost distributed multi-tile wall.

design, explaining in detail the software used and the system architecture as well as different physical setups and interactive research case studies.

3.1 Related Work

Game engines have been adopted extensively as tools to solve and convey research in many fields due to their unique characteristics with increased interactivity and compelling graphics performance. Example of successful research, projects and development directions using game engines for research can be found in [196, 248, 223, 29, 289]. This situation had led several researchers to explore the use of game engines

to support high-end VR, in particular immersive displays.

In the following paragraphs a related work overview focused on high-end Immersive VR systems based on modern Graphics Engine is provided.

Rajlich's CAVE [278], developed at NCSA, is probably the first immersive implementation of a popular computer game Quake II . It has been followed by CAVE Quake III Arena, based on the open source Aftershock engine. Jacobson et al. [145, 146] proposed the first system implementing low-cost VR with CaveUT based on Unreal Tournament 2004, however no porting to a newer platform seems to be planned. A similar system has been proposed by [307] using Half-Life 2 game engine. This system provides the full potential of the game engine but it is designed to be running in a simple screen setup with no distribution. Juarez et al. [151] proposed a distributed environment based on CryEngine 2. Their solution allows the creation of a low-cost multi projection Cave System with single processing unit but, as a closed solution, it does not provide cross-platform portability or tools for networked collaboration. Moreover, they have reported average frame rates less than 20 fps, which may not be sufficient to support a comfortable viewing and interaction experience. BlenderCave [94, 269] is the VR extension of the open-source Blender engine but its VR version demonstrated limited rendering performances, without support for dedicated I/O VR peripherals.

More Recently, MiddleVR [141] was released as a middleware extension for Unity 3D [355]. MiddleVR is at the present a successful tool providing support for generic hardware configurations and the tools of a commercial high-level graphics engine. However, it is a closed commercial solution relying on licensing agreements for development and distribution and its visual performances do not reach that of the most advanced commercial game engines. Kreylos [174] proposed the Vrui toolkit for interactive and high performance VR applications. The system is scalable, generic and provides a module for the development of a distributed environment. However, the framework is not integrated with state of the art engines providing advanced rendering effects, character animation tools, and integration with modeling packages. From the same framework tele-immersive systems with focus on high-quality user appearance and immersive collaborative applications have also been developed [387, 285]. These systems are based on stereo-cameras generating high-quality 3D images or triangulated meshes of the user that are streamed between participants. These systems are specialized for their applications and are also not suitable for multi-user connections due the high bandwidth required.

Finally, CaveUDK [208] presented a high-level VR middleware based on the commercial game engine Unreal Engine 3.0 (UE3). The system relies on a distributed

architecture reinforced by specific replication patterns to synchronize the user's point of view and interactions within a multi-screen installation. The system is flexible and can be extended to different hardware configurations but the programmer has to take into account aspects of data synchronization and object replication. The engine also requires monthly subscription or license agreements.

3.2 System Architecture

Four main principles have guided the design of this Immersive Virtual Reality framework:

High level abstraction The ability to automatically handle and hide to the application developer the low-level communication and hardware-dependent calls, which are needed to maintain networked clusters and their internal rendering mechanisms synchronized;

Collaborative sessions The ability to connect multiple application instances in collaborative virtual sessions, with support for tracking and controlling avatars and autonomous virtual characters in the shared virtual environment;

Scalability The ability to scale from a single desktop computer to more complex distributed systems with varied tracking devices, 3D visualization technologies and multi-platform requirements;

Cross-platform and open code The framework combines and orchestrates different open-source (or open-code with no use restriction for educational and non-commercial usage) and multi-platform (with the exclusion of optional platform dependent device driver modules) libraries.

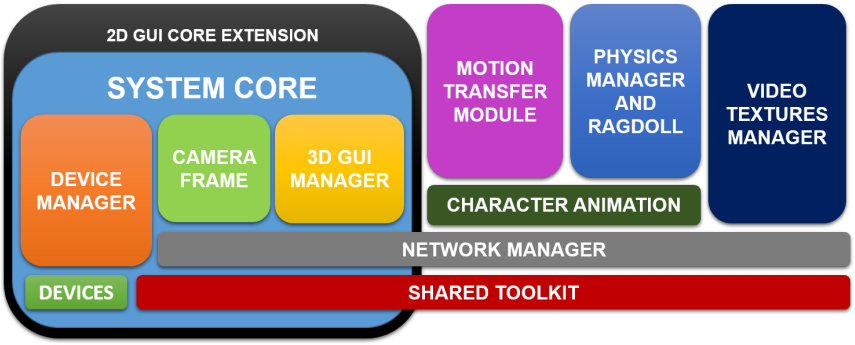


Figure 3.2: Overall system architecture.

The framework is implemented in C++ and its main entry point is a derivable core class. Depending on the functionality required by the application, different modules instantiated as managers can be employed (see figure 3.2). The system core and its modules are highly flexible and the main functionality of each manager can also be customized from a unified configuration file.

Each module and the core class were modeled using a distributed approach and designed to achieve transparent data replication to help the user develop optional cluster based applications. All of them are designed on top of the network distribution manager. The internal data communication adopted by the manager (for distributed rendering) follows the Master-Slave (MS) communication mechanism with data replication (design pattern and communication models are explained in detail in section 3.2.3 and 3.2.4).

The framework has been designed to adapt to the type of hardware selected and the same executable, joining a collaborative session or running in distributed rendering cluster, configures itself automatically.

In the next paragraphs the framework is described thoroughly exposing each relevant component architecture and features.

System dependencies

The VR framework developed is a complex system with a broad adaptability to different hardware and configurations. Several external low level libraries are orchestrated by the framework. These libraries have been chosen for their adaptability, portability to different platforms and for their unique features compared to similar open code libraries.

The rendering engine is based on the OGRE Game Engine [345], which provides a wide variety of tools and high-end graphics capabilities. It also supports powerful scene exporters from all major 3D modelers.

The low-level network communication layer is based on the cross-platform game networking engine RakNet [324]. RakNet was chosen since it is designed to be a high performance, easy to integrate, and complete solution for games and other applications. RakNet provides a versatile and unified development environment allowing the easy access and extension of the low-level UDP datagram protocol communication (with the possibility to optionally use reliability, message ordering and sequencing on multiple channels, automatic congestion control, message coalescence, splitting and reassembly of packets and object serialization and deserialization) and it also provides advanced functionalities for the development of networked, large scale and

reliable communication (designed for reliable WAN communication over the internet). The high level components included and used for several projects consists in: game object serialization, creation and destruction; NAT Punch-through; Remote Procedure calls, etc.

The Graphsim toolkit [155] provides management of motion capture animations, flexible skeleton classes supporting varied joint parameterizations, animation blending techniques, and diverse character animation algorithms such as for motion planning and analytical inverse kinematics with collision avoidance.

The following libraries are also employed: Bullet Physics [344] to achieve Real-Time physics simulation and Physics based character animation; Qt [81] and FLTK [88] for embedded and stand alone 2D GUI interfaces; TinyXML [109] for XML configuration file parsing; LibVLC [262] for animated 3D textures and video streaming; WiiMote Lib [101] for interface with Wii-Mote controllers; ViconSDK [361] for the integration with Vicon Blade and Vicon Tracker motion capture software; 5DT SDKs [1] for hand tracking with data-gloves; and the Microsoft Kinect SDKs [224] to enable markerless motion capture and gesture recognition through the Microsoft Kinect Sensor.

The platform-dependent tools are driven by an external application included in the framework (described in Section 3.2.4).

3.2.1 System Core

The application core is a monolithic entity that orchestrates the requested managers, maintaining the shared states, handling the user's interaction events and providing to the developer a high-level abstraction of the system.

The framework application flow is characterized by four major steps extending the common OGRE rendering cycle: the pre-configuration stage; the engine configuration stage; the post-configuration stage, and the main application loop. The first three phases are also referred as application *bootstrap*. Every application step is exposed to the developer with derivable methods that can be treated as access points to customize each application stage.

During the bootstrap process, the framework initializes all requested modules configuring them according to the cluster physical layout, loading and pre-processing all the requested local resources. During this stage the core determines if an application instance will be running as an active participant into a shared communication (active participant into a shared session) or, in case of distributed rendering, if an application instance is a slave instance waiting to be directed by another entity (for

example, each applications running in a distributed rendering cluster are driven by a master application).

Following the bootstrap is the main rendering cycle. The rendering loop follows the standard update states, render scene, swap buffers mechanics employed by a common 3D accelerated real-time rendering platform. In case of distributed rendering, the master instance drives the rendering by enforcing a synchronized swap step between each node involved in the rendering process. The rendering step is always preceded by enforced shared objects data synchronization (see Section 3.2.3 for details).

The core class has been designed to be adaptable to several type of application deployment. Each active core instance can be deployed as standalone application where the window generated is fully controlled by the core itself (this is common deployment scenario used in video games where the main windows preferably runs in full-screen mode with optional in exclusive input control) or integrated with advanced user interface system, such as Qt [81] or Fltk [88].

The core class, as previously mentioned, directly depends on four main components: the camera frame, the device manager, the 3D GUI system and the network manager.

3.2.2 Camera Frame

The camera frame module is responsible for creating the physical rendering window array and for performing generic virtual camera manipulations. Conveying depth and layout is a key element of a VR system. In a multiview projection system with a strong user centric paradigm such as a CAVE, a powerwall or more genetically in any user-perspective stereo vision system, egocentric depth perception is simulated using binocular stereopsis and motion parallax. Creating egocentric depth perception in a multi-screen environment using a distributed architecture requires the implementation of a specific camera system.

The camera frame is based on two main components: the virtual reference camera and the user's point of view. The reference camera (RC) provides generic distributed camera control with respect to the virtual scene. The user's point of view (PoV) component is responsible for achieving user-perspective stereo rendering. PoV describes the physical position of the user with respect to the projection system selected and it is then expressed in RC coordinates. PoV maps accurately the topology and geometry of each projection system involved in the rendering and the position of the user with respect to the virtual scene (PoV is hierarchically connected to RC) to

achieve off-axis stereo projections. The Figures in 3.3 show the off-axis stereo pair image generation with respect to the user point of view and an example of multi-pipe or tiled projection systems. In the presence of distributed rendering the RC and PoV are streamed to each passive rendering client and each rendering node uses this information to generate images accordingly.

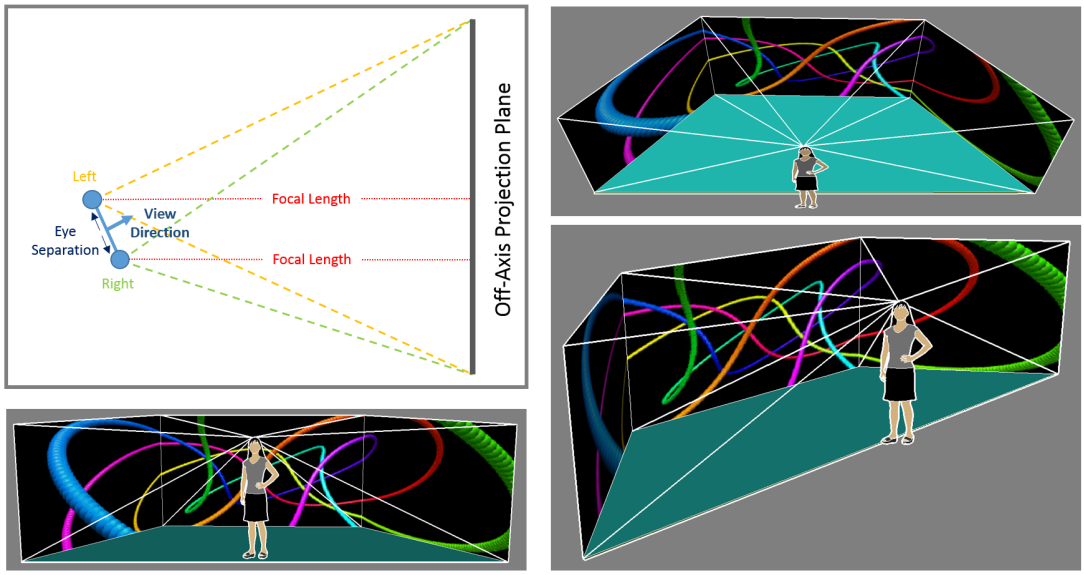


Figure 3.3: Off-axis projection to create stereo pairs, note the non symmetric camera frustums generated.

If the PoV of the user is associated with an input device, acting as a head-tracker, the user will be able to perceive the scene in a metrically correct fashion (user-perspective stereo vision). When the PoV of the user is updated in real-time from the head-tracking device, only the primary user will perceive the scene correctly and extra users looking at the virtual scene, if present, will likely incur in motion sickness and discomfort since the images produced will not be dependent on their specific Point of Views. In this modality a hand-controller navigator is used to change the position of the RC allowing navigation operations in virtual spaces larger than the physical walking space where the user is tracked. In order to support a minimal safety mechanism to prevent accidental damage to the projection surface a proximity guard control, to warn the user if its input devices get too close to the screen, is included in the camera frame as a built-in warning.

The system can be configured to achieve multi user stereo vision, similarly on

what is achieved in commercial 3D TVs or in 3D movie theaters, by fixing the PoV to a specific position with respect to the screen. In this configuration, in order to accommodate people with different heights, the PoV is placed, by default, at the center of the virtual screen and at a perpendicular distance of $2.5m$ from it. This distance and position can be adjusted depending on the audience and the experience requested by the application.

In all the visualization configurations, the reference frame (RC) is manipulated by camera manipulators that can be optionally linked to the hand-controlled navigator. These manipulators are customizable entities that can be attached and interchanged as needed.

In order to support a wide range of configurations but still keeping a high level of abstraction from the projection system employed the *virtual screen, window and viewport* model was developed. A virtual screen defines the physical surface where the 3D scene is projected (the surface can be composed by multiple hardware screens or projection surfaces). Parameters like dimension, margins, position and orientation (with respect to user working space) can be specified (as shown in Figure 3.4). A virtual window is identified with an operating system's window with a graphics pipeline associated with it (distributed or local). Several windows can be associated per screen and they can be designated to render any specific rendering nodes (for example in case of CAVEs or multi-tiled walls). Sub-region rendering can also be assigned to create multi-tiled projections. Finally, a virtual viewport is assigned to a camera and it defines a rectangular viewing region. Any number of viewports can be created inside a window however, when the system is configured for stereoscopic rendering the specific number of needed viewports is enforced due to physical requirements (e.g. 3DTv stereo vision transmission protocol requires a single window with two split viewports, etc.).

The screen/window/viewport model can describe a wide range of VR environments. The framework has been tested extensively with passive stereo vision systems with overlapped windows or dual viewport windows (with multi-cluster synchronization or multi-pipe) as well as 3DTvs (active and passive) and NVidia 3D vision active stereo [252]. At the present, the system does not support multi-tiled/distributed active stereo vision, due to lack of testing hardware, but the system can be easily extended to support this case.

The Camera Frame system also includes a 2D overlay layer designed to level a multi-pipe rendering surface brightness contribution uniformly or through texture alpha blending.

```
##########################################
# Frame Configuration
#
section frame
    defaultuserpos 0.0 1.75 3.0
    framebgcolor 0.0 0.0 0.0 1.0
    framenearfar 0.03 1000.0
    framepos 0.0 0.0 0.0
    framelookat 0.0 0.0 -1.0

    activeeyesep 0.0675
    activescreens screenmastersimple

    passiveeyesep 0.0675
    passivescreens screenpwall

    brcontrolmaterial core/brighness_control_20
    scproximitymaterial core/proximity_control_50

    subsection screen screenmastersimple
        mode mono
        position 0.0 0.0 0.0
        normal 0.0 0.0 1.0
        dimensions 4.57 2.25
        margins 0.13 0.13 0.13 0.13
        usewindows winmastersimple
    subsectionend
```

```
subsection screen screenpwall
    mode stereo_dual
    position 0.0 0.0 0.0
    normal 0.0 0.0 1.0
    dimensions 4.57 2.25
    margins 0.13 0.13 0.13 0.13
    usewindows winpwalltile0left winpwalltile0right \
               winpwalltile1left winpwalltile1right \
               winpwalltile2left winpwalltile2right \
               winpwalltile3left winpwalltile3right \
               winpwalltile4left winpwalltile4right \
               winpwalltile5left winpwalltile5right
subsectionend

subsection window winpwalltile0left
    hostname pwall-rackm-0
    fullscreen yes
    fullscrres 1024 768
    resolution 320 200
    position 0 0
    useviewports vptile0
subsectionend

subsection window winpwalltile0right
    hostname pwall-rackm-0
    fullscreen yes
    fullscrres 1024 768
```

Figure 3.4: Camera Frame configuration file extract. This sample code shows the use of the screen/window/viewport model setup for a distributed rendering cluster with 6 rendering tiles using passive stereo vision.

3.2.3 Network manager

The network manager and its structures are key components of the framework architecture and the majority of the modules use them extensively to orchestrate and maintain distributed and replicated structures between application instances. The network manager has been designed to accommodate different types and custom made communication engines that can be selected according to specific application or system needs.

By default, the framework provides two communication engines. The first engine is highly specialized to support internal cluster communication and distributed rendering (master-slave interactions). The second engine is designed to support a general purpose communication channel between collaborative application instances (Collaborative Virtual Environments) over the internet (master-master interaction).

The internal communication engine internally defines two communication layer abstractions. The first layer is based on a low-level message passing mechanism and it is designed for the internal core development. This layer is responsible to drive

internal core event messages, to maintain the core's shared states and to maintain the internal nodes rendering synchronization.

As previously mentioned, this engine (as well as the general purpose communication engine) is based on the game networking engine RakNet. The choice of using RakNet, as a unified communication platform, instead of using the bare bones UDP or TCP protocols, was driven by the wide flexibility of the features provided by the library. The main communication protocol used by RakNet is based on UDP, for fast replication of messages through datagrams multicast, and its capability of providing communication channels with independent message sequencing, ordering and reliability to support the different type of messages and objects to be serialized, streamed and deserialized. These features simplified the development of the system and the maintenance of coherent states obviously adding some minimal performance cost. An extended discussion on performances and trade off about these choices are presented later in this section and also in the Results section 3.4.

When the framework is configure to support distributed rendering the internal communication engine is initialized. During this initialization process the engine takes control of the application core rendering cycle enforcing synchronization between the master application and all the slave nodes participating in the communication. This last step is performed enforcing a sliding window process with mutual waits between master and slaves during key specific phases. As depicted in Figure 3.5, the master node starts broadcasting the high level application states to each participant. After this stage the master application enters in a wait state until it receives an acknowledgment of completed update and after each rendering phase from the clients. When the ready state is achieved the master can signal to each participant to swap their internal buffers and render on screen synchronously.

The system enforces software synchronization between the node since the cluster is designed to not be dependent on a specific hardware solution, adopting commodity and inexpensive solutions.

In our testing environment, the software implementation of the sliding window synchronization, without considering the core rendering cost, requires around $3.5ms$ (std.dev. $\pm 1.25ms$ on a 7 nodes distributed rendering cluster). The use of a dedicated software solution to enforce Networked Time Synchronization and Message Passing Interface (MPI) [159] (or the use of hardware based solutions for distributed clock synchronization such as NI-Sync with IEEE 1588 Precision Time protocols [143]) could be adopted for a strict enforcement of synchronized render calls achieving a synchronization time on the order of 10μs. Conversely, the adoption of such synchronization mechanism alone will not be enough to also avoid a possible video

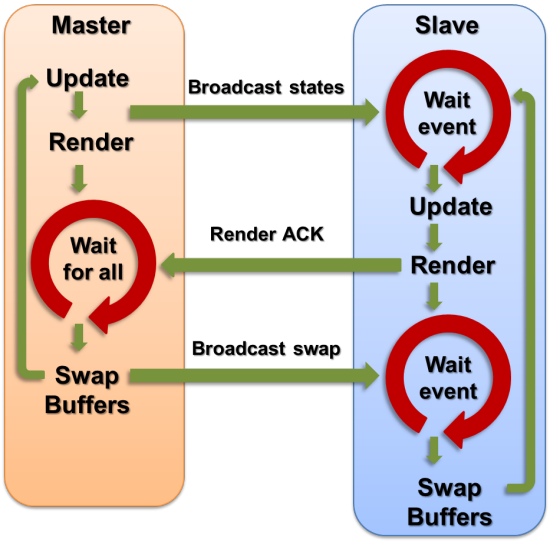


Figure 3.5: Rendering cycle synchronization.

desynchronization, between different nodes in the cluster, during the swap phase. The Swap phase is heavily dependent on the internal computational stress that each graphics card undergoes during each frame generation (number of triangles displayed, shaders, effects applied, post-processing, anti-aliasing and filtering etc.). A common solution is to adopt a dedicated hardware solution designed to synchronize the internal graphics card rendering swap phase (such as Nvidia Frame/Gen Lock protocols [254, 253]) or adopt a high-end graphics card that natively support this feature. The use of such a system would ensure the removal of the software sliding window synchronization step.

The second layer of communication has been designed with the goal to simplify the creation, maintenance and update of user defined networked objects. This layer is responsible to accept specialized and self-maintained objects called *communication channels*. A communication channel is a derivable class that automatically replicates, destroys and updates itself in every peer participating in the communication. The communication channel concept extends the distributed object replica pattern enforcing the update of each object during each rendering cycle's update. The communication channel ensures one-way flow synchronization from the source objects to the each replica object. The communication channel also implements an internal object remote procedure call mechanism (*channel events*) that can be enforced at a time of the call or it can be delayed at the time of the global update.

The base communication channel class is an abstract class that requires the developer to complete specific class methods triggered during specific object's life stages. The initialization and update stages are mandatory and they both need to be customized by the user. The initialization stage is triggered after a new communication channel is created (both by the source object and after the distributed creation request of each replica). The update stage is triggered during each application update stage (this stage is ensured to have all the internal states synchronized with the source of the channel according to each channel streaming policy).

The distribution of the internal stage of a channel happens in three phases: the construction, the update and the destruction serialization. These phases have been separated since the serialized data in these stages and the transmission policies can be different (for example the datagram sequencing or the ordering enforcement). Figure 3.6 depicts the three stages and the communication between the source object and the replicas in detail.

Since the serialization and deserialization of data can be a complex and tedious development step (the developer needs to know low-level details such as: the objects's serialization structures and how to stream it; who is transmitting and who is receiving; providing mechanisms to save bandwidth and so on) the channel class has been enhanced with the optional use of *autostreams*. The autostream class is a templetized structure that, similarly to how the Reference Pointers pattern works, is able to maintain any kind of class or data type but allowing the developer to access the class in the same way it would access the original class object. The autostream main duty consists on extending each stored object to be serialized through a channel and automatically be updated back during the channel deserialization. Another advantage of the autostream class is the maintenance of an internal state history that will prevent the class from being streamed if its state is not changed.

The autostream pattern is handled by a channel through a registration/unregistration mechanism. If the channel internal member variables are all handled by the autostream pattern the serialization and deserialization mechanism is completely hidden to the developer extremely simplifying the distributed object's development and maintenance.

The second engine provided by the system is the external communication engine. This engine handles object replication between peer applications implementing the standard distributed object replica pattern provided by the RakNet. The external engine is designed to be fully reliable in WAN connections and it can be optionally extended with network compression, security layers and NAT punch-through. In addition, the external communication engine is only designed to maintain data

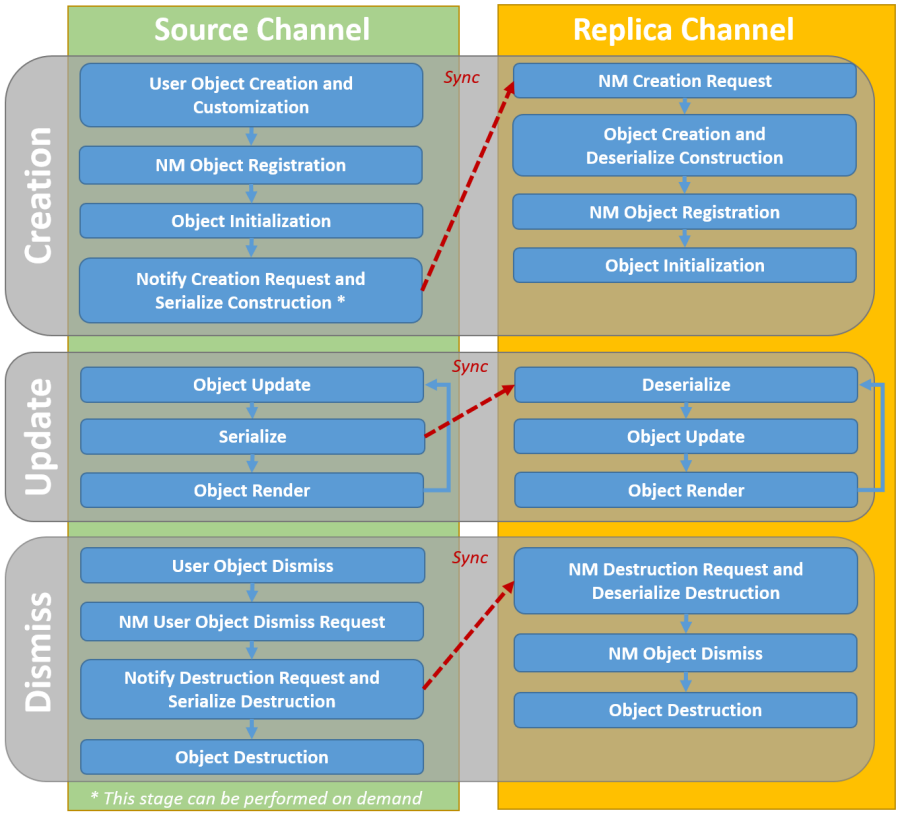


Figure 3.6: This figure depicts the *communication channel* stages for the object construction, destruction and update. On the left the flow of the source channel object is shown. On the right the figures show the flow of each replicated object. NM is the Internal Network Engine.

synchronization between instances and does not affect the rendering cycle.

Similarly to the internal communication engine that uses *communication channel* objects distributed objects used in the external engine has to extend the *communication replica* class. After an object has been created by a peer the object is replicated in each peer being part of the shared session. Modification of an object is done by requesting changes to the object owner. Concurrency control follows the optimistic scheme [114] allowing users to update objects without conflict checks and thus conflicts can occur. *Autostream* pattern is also supported by the external communication manager.

3.2.4 Device Manager, Video Texture Manager and 3D GUI

The device manager provides a simple mechanism to allocate interaction devices. When an input device is requested a virtual device is created and its device-dependent messages are translated into a virtualized higher level format.

The framework natively supports mice, keyboards and joysticks. A client-server extension has been developed to retrieve information from platform-dependent devices, such as: Blade's motion capture streaming, Vicon Tracker's rigid object streaming, data-gloves, 3D-Connexion mice, etc. For any needed device the device manager is connected to a networked service (running in the machine controlling the device) to retrieve the device data. Virtual devices can be optionally assigned to the camera manipulators, bound to virtual pointers or assigned as head tracking entities.

Virtual pointers are interaction objects that allow the user to interact with the scene: select objects, trigger events and spatially manipulate objects. Virtual pointers implement a ray casting mechanism (rays are shot from the position and orientation of the pointer in space) performed per object or at the object geometry level. Objects can also be registered to raise events, for example: pointer in/over, release, select, attach, etc. The device-pointer binding is a flexible system that decouples buttons from motion controllers, allowing different combinations of input devices.

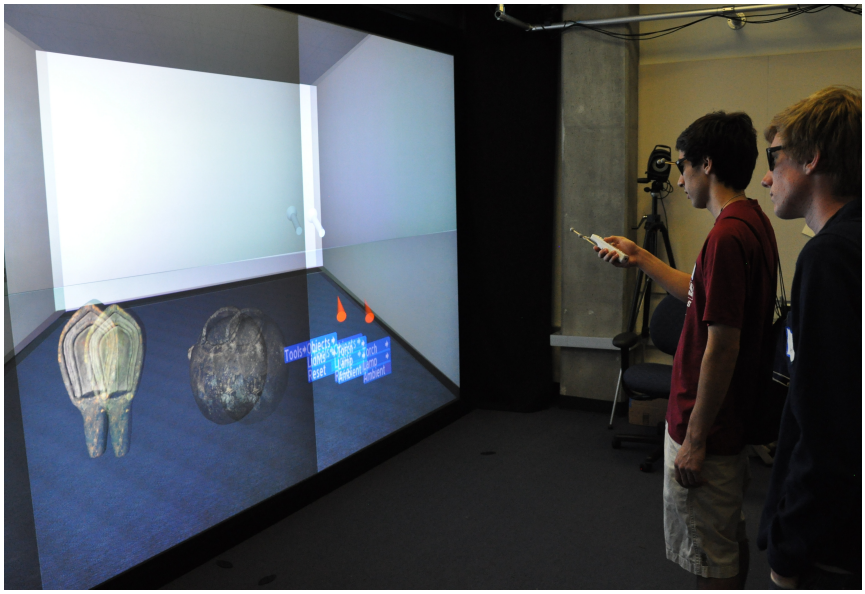


Figure 3.7: Example of objects manipulation through the virtual pointer assigned to the hand-controlled device.

The 3D GUI manager includes primitives for distributed interactive widgets. Currently the system supports specialized widgets like buttons, labels, spin-boxes and sliders. Widgets can be hierarchically arranged inside floating panels or menus, in a tree-structured manner. The coexistence of widgets and pointers grants to the user a full scale calibrated immersive interaction model. The framework also provides advanced pre-built panels, such as for recording and playing captured animations on-line.

The framework also includes synchronized video texture streaming. Supported formats depend on the codecs installed in the machine, the synchronization between nodes is achieved through a play rate recovery algorithm. A video stream can be also mapped in video planes widget and incorporated inside 3D GUI if needed.

3.2.5 Character Animation and Motion Reconstruction

Character animation and motion reconstruction modules provide high level functions to enable the design of virtual reality applications based on characters with real-time animation transfer directly from user's body postures. Both modules are extending the main framework's *character* class. That character class allows the loading of high-resolution skinned models and it is designed to maintain efficiently each character's property over distributed application's instances. Different formats for key-frame animation can be loaded and animations can be interpolated in a distributed way (Animation-Driven), or synthesized at the master node and streamed as synchronized joint values varying over time (Joint-Driven). The character networked synchronization is bandwidth efficient and supports automatic determination of minimal updates.

The character animation module includes 3D widgets to interact with full-body character animations. The system includes tools to collect, play, record and store single motions; as well to trim and annotate motions to be used by diverse motion synthesis applications. The animation module also includes tools to compute and visualize on-line information from motions being tracked, for example: angles between limbs, end-effector trajectories, and joint range of motion (see Chapter 4).

The motion reconstruction module allows real-time full-body tracking and retargeting to virtual characters of varied dimensions. The module works coupled with the device manager querying the specific input devices being used. The motion tracking process can be performed with full-body tracking (through Microsoft Kinect, inertial sensors or Vicon cameras). A motion retargeting calibration and real-time mapping algorithm from a reduced set of sensors is available based on in-

verse kinematics. Finger motions can also be reconstructed from data-gloves. The reconstruction algorithms follow the solutions discussed in Chapter 6.

3.3 Applications Using the Full-Body Interface

Our framework has been successfully employed on a number of research projects (see figure 3.1). Several of the projects were able to explore new immersive full-body interfaces and interactions with avatars and virtual characters. Examples are given below.

Motion Modeling This application focuses on training scenarios where the user (expert in the training subject) is able to demonstrate to a virtual human how to perform parameterized motions. Later, during the training phase, the virtual human is able to reproduce the motions in interactive training sessions with apprentice users learning the training subject. The system includes new algorithmic solutions for motion parameterization and an intuitive motion interface for enabling the direct demonstration of parameterized actions and gestures by demonstration (refer to Chapter 4 for a detailed description).

Physical Therapy The presented framework is being employed to explore new VR approaches to improve the delivery of upper-body physical therapy. The overall process of Adaptive Physical Therapy is explained in details in Chapter 6. The overall application is being implemented in two possible configurations: a low-cost hardware solution based on Kinect (suitable for home or clinical setups); and a high-end solution based on a Powerwall immersive VR system. The application allows therapists to create new exercises by direct demonstration, and it then provides automatic interactive exercise delivery and monitoring with an autonomous virtual therapist. The application also supports networked collaboration between the therapist and the patient, enabling the possibility of remote consultations .

Interactive Motion Visualization A full-scale visualization tool has also been implemented for inspection of the results computed by a full-body motion planning algorithm [134]. Motions are synthesized from a motion graph algorithm in an environment with obstacles, and then integrated with a motion planner designed to compute upper-body motions for given tasks. Computed motions can be visualized in full-scale, and the system also allows interactive adjustment of the underlying motion databases available for the motion graph. Powerful interactions are enabled with the possible immersive manipulation of the environment, allowing users to closely evaluate the capabilities and limitations of the involved planning algorithms.

Additional Projects Additional applications supported by the framework include: visualization and interaction application for high-resolution laser scanner data, cloud of points and virtual reconstructions; 3D archeological fieldwork video player; visualization of archaeological environments with interactive content navigation and display [157, 192, 79]; studies on perception and user interaction; virtual exhibitions; and tools for virtual teaching and virtual presentations. The framework has also been successfully used by students for development of diverse class projects (for example: physics-based particle simulation project with user perspective-interaction, etc).

3.4 Results

The framework presented has been fine-tuned, tested and evaluated in different physical configurations, from simple desktop computers to distributed clustered environments.

The primary test platform for our immersive virtual reality setup is a six-tile Powerwall arranged in two rows of three panels, and integrated with a 10-camera Vicon motion capture system. The Powerwall is retro-projected (4.56m by 2.25m) and consists of seven commodity Linux-based rendering nodes (Pentium Q9550 2.83GHz GeForce GTX 280 4Gb RAM), twelve projectors with circular passive polarization filters (1024x768px resolution each), an external windows device server, and a Gigabit Ethernet. We also performed experiments simulating CAVE-like environments (from the standard 4-wall configurations to more customized multi-tile screen compositions).

The Powerwall has showed to be well suited for achieving high-resolution in wide working spaces, for example to accommodate several visitors or when manipulation of large scenes is needed. As a comparison, CAVEs provide immersive experiences covering a larger field of view, which minimizes the problem of suspending 3D immersion when the user visualizes content near screen borders. The system is compatible with the NVidia 3D vision system (DirectX mode), and has also been tested extensively in a desktop setup with a 3D TV monitor. In addition to Vicon cameras, we have also experimented with OptiTrack for tracking user motions and the user-perspective stereo vision. Both systems are based on optical infra-red cameras tracking reflective markers. When conditions are not ideal both suffer from marker occlusion and camera-marker proximity problems. The probability of such problems to occur is increased when the capture space is overly small, as in installations in small rooms.

To achieve an immersive visualization system that is effective it is essential to guarantee good synchronization and calibration of user tracking, rendering nodes,

and virtual scene scaling with respect to the user workspace. A system calibration error on the order of millimeters is enough for an uncomfortable and ineffective 3D immersion. Moreover, low latency is a critical feature of a VR system, as it is not only a negative factor in simulator sickness, but also considerably affects interaction [346]. Consequently, VR systems should support both high rendering refresh rates ($>= 30Hz$) and high interaction responsiveness ($<= 150ms$ for digital games [150]).

Table 3.1: System benchmarking in the 7 nodes distributed rendering cluster. The table shows the application frame rate (FPS), the triangle count, the communication channels opened (ch.), averaged rendering time (t_r), averaged swap time (t_s) and averaged network time (t_n) while progressively loading low bandwidth consumption Animation-Driven characters (no.).

no.	**tri. (K)**	**ch.**	**FPS**	t_r **(ms)**	t_s **(ms)**	t_n **(ms)**
15	69	42	158	3	1	2.33
30	130	57	132	3	2	2.25
45	200	72	125	3	3	2.75
60	255	87	108	5	2	2.26
75	315	102	92	5	3	2.87
90	370	117	83	6	4	2.04
105	439	132	75	8	3	2.33
120	501	147	68	9	3	2.71
135	563	162	62	11	3	2.13
150	630	177	57	11	4	2.54
165	687	192	52	13	4	2.23
180	750	207	48	14	4	2.83
195	810	222	46	15	4	2.73
210	872	237	41	16	5	3.39
225	934	252	39	17	6	2.64
240	998	267	37	17	7	3.02

Performance experiments were measured to isolate the effectiveness of the channel state history bandwidth saving mechanism and preservation of the frames per second (FPS) rate.

The first test application developed loads low resolution Animation-Driven characters (15K vertices, one 1Kx1K texture, 20 joints) with software skinning and the character animation is synchronized through time and location only (6DOFs in root joint and 1 real value expressing the animation time). This test is designed to isolate the actual system performance during increasing rendering stages with scene updates and low bandwidth consumption. Table 3.1 reports the data sampled during our trial. Fifteen new characters are loaded in the system every 5 seconds. The *autostream* mechanism for this trial has been disabled so the channel's internal communication structure and the channel user data are always retransmitted to leverage

the serialization construction and update serialization stage. The frame rate and time sampling are averaged during this period of time.

From the table we can clearly see that the frame rate, rendering time and rendering buffers' swap time massively depend on the number of characters spawned while the network transmission and synchronization time is stable and does not affect the rendering time (see also Figures 3.8). The introduction of new characters' channel structures does not significantly affect the rendering performances. With the introduction in the scene of 240 characters (1 million triangles mapped in software skinned skeleton structure) the cluster rendering still maintains an overall frame rate of about 37 frames per second.

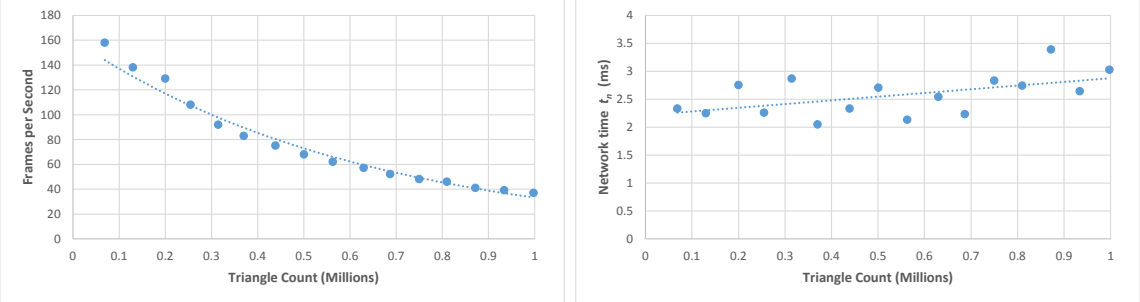


Figure 3.8: System benchmark: low bandwidth application. On the left: application frame rate progressively decreases during the introduction of new characters. Each dot represents the introduction of 15 new characters in the scene. On the right: network time during a distributed rendering cycle. The introduction of new characters' channel structures does not significantly affect the rendering performances.

The second test application loads high resolution characters (100K vertices, seven 2Kx2K pixels textures, 67 joints), software skinning and per joint data transmission (6DOFs in root joint and 3DOFs in other joints). A new character is loaded every 5 seconds and the application is benchmarked for 150 seconds. The tested motion did not affect the finger joints, which represent 70% of the joints. Four types of conditions where tested and compared: the system had all the saving bandwidth mechanisms enabled and running in a normal condition; the system running, again, with all the saving mechanisms enable but without displaying the characters on screen (the only extra effort required to the system besides the usual synchronization step consisted of each characters' skeleton update stage); the application running with the channels *autostream* disabled; and the character's joint history mechanism disabled.

During this test we tracked information such as: bandwidth consumption (bytes

transmitted from the server to each node), the cluster frame rate, triangles rendered on screen and so on. Figure 3.9, on the left, shows that the system FPS is preserved when both bandwidth preservation mechanisms are enabled. Obviously, the case where the characters were not rendered results in a higher application frame rate. Moreover, as depicted also in Figure 3.10, it is noticeable a decreasing trend in performance of around 1.5ms due to a higher update stage computational cost.

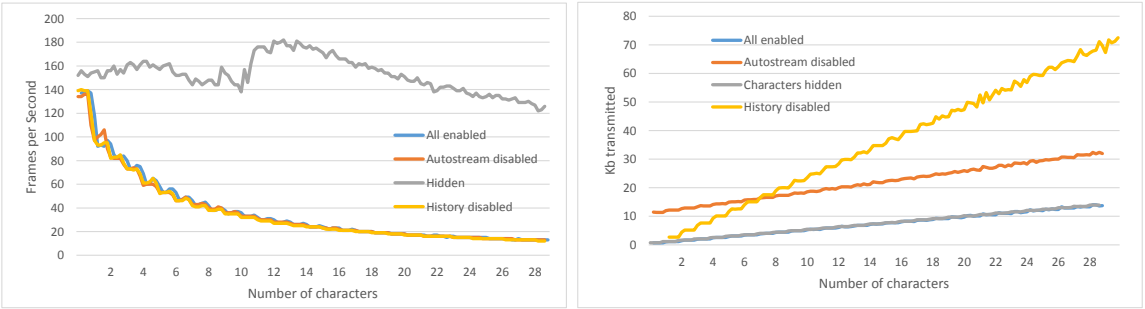


Figure 3.9: System benchmark: high demand application. In both figures four variations are reported: all saving bandwidth mechanisms are enabled; the channel *autostreams* are disabled; the character's joint history is disabled; and the saving mechanism are enabled but the characters are not displayed on screen. On the left: application frame rate progressively decreases during the introduction of new characters. On the right: geometry introduced in the screen and bandwidth consumed.

Looking at the bandwidth usage, the *autostream* mechanism significantly decreases the amount of data transferred in a rendering cycle. This is explained by the fact that most of the data transferred in channels are used as object bootstrap that will not be modified very often during the lifetime of the object. Considering the character history mechanism the total bandwidth was reduced by almost eight times per character animation. Since in our example the 70% of the joints are not touched by the animation the real saved bandwidth using this technique is around 13%. The technique is most effective in situations where many states remain static after the initial object update. Even if all the states are updated at each frame, the overhead of the bandwidth saving mechanism is minimal and the overall system performance and FPS is not altered. These results are shown in the Figure 3.9 on the right. The internal data transmission can be further improved enabling data compression, however slightly degrading CPU performance.

In terms of interaction responsiveness (navigation and firing-buttons events) our system introduces delays times in the order of few milliseconds, for devices directly connected to the main rendering node (such as keyboard, mouse, joystick etc.) to

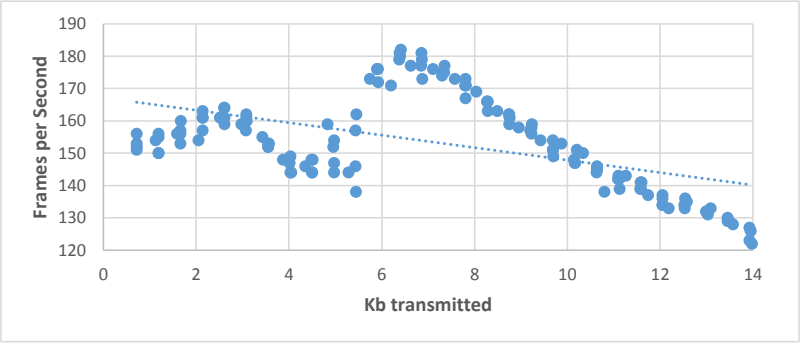


Figure 3.10: System benchmark: characters not displayed highlight. A decreasing trend in frame rate performance is shown due to higher update stage computational cost.

up to $40 - 50ms$ for devices connected to the networked device manager (such as gyro sensor, wii-mote controllers etc.). The bottleneck of out configuration consists in the use of Vicon Blade as our main tracking device for rigid objects with respect to the virtual screen. The tracked objects can be used for different purposes but in our testing setups mainly to track user's head position (and generate off-axis projection) and directly mapped to the virtual pointers. The use of Vicon Blade version 1.7 introduces in the system a quite noticeable lag that slows down the interaction responsiveness of these two components of about $250ms$, slightly introducing small discomfort in the 3D perception during fast interaction or suddenly change of point of view. The system has been briefly tested with tracking software optimized for real-time rigid object such as Vicon Tracker and with the employment of different motion tracking solution such as OptiTrack. The responsiveness time in these cases decreases safely under $150ms$.

Overall, the performance results of our system demonstrated the appropriateness of our VR framework implementation regarding rendering performance and end-to-end latency. Moreover, comparing our system with similar solutions, for example the Unreal Tournament VR framework proposed by Lugrin et al. [208], we can conclude that we achieved similar results. Their solution has been tested with a faster machine and dedicated graphics hardware with GenLocking two generations newer than our commodity solution. Considering the benchmark comparison produced by PassMark/G3D Mark [325] the polygonal processing power of their solution is about four times faster than ours. Our framework solution maintains on average 54fps using the interactive model visualization application with around 1 Million visible polygons displayed (total cluster time $18ms$ - $4ms$ render - $11ms$ swap time). Their

solution was reported to maintain around 45-50fps with between 1.5 to 2.5 Million visible triangles in the rendering scene.

3.5 Conclusion

This chapter presented a new software solution for the development of immersive collaborative virtual reality applications with user-perspective stereoscopic rendering and interactive full-body characters. The major strengths of the system are its integration to a modern and complete graphics rendering engine, its integration with high-performance toolkits with source code available (or open-source) that guarantees simplified maintenance and upgrades, and its versatility to scale well to different hardware configurations.

The framework handles most of the distributed programming and collaborative VR development difficulties thanks to automatic streaming channels and replica mechanisms. Moreover, the framework includes high-end animation and user motion reconstruction tools that sets our system apart from the other open VR frameworks. The number and diversification of the supported applications demonstrate the effectiveness of the proposed framework.

CHAPTER 4

Motion Modeling by Direct Demonstrations

A central goal in the area of autonomous virtual humans is to achieve virtual assistants that can effectively interact, learn, train, and assist people in a variety of tasks. We focus on the particular problem of modeling motions for interactive training applications requiring complex gestures and actions to be reproduced realistically and with precise parameterizations with respect to spatial constraints in the environment.

Modeling and parameterization of realistic motions is clearly an important problem in a wide range of applications involving virtual humans. One approach for achieving precise parameterizations is to rely on algorithmically synthesized actions and gestures [249, 169], however it remains difficult to achieve realistic full-body results and a specific computational model is needed for every action to be simulated. Another important limitation of algorithmic approaches in many training applications is that the motions to be reproduced may only be known by experts in the subject area of the training. Such cases are clearly better handled by motion modeling solutions based on motion capture.

Several systems based on motion captured (or hand-crafted) animations have been developed and are able to achieve highly realistic results [96, 348]. However the process of building the set of needed motions for each given scenario is often time–consuming, and it remains difficult to precisely parameterize pre-defined animations with respect to spatial constraints.

This chapter proposes an interactive motion modeling framework for addressing these many difficulties in an integrated fashion. The framework is designed to be used in two distinct phases: in the **modeling phase** the user demonstrates to the virtual human how to perform parameterized motions, such that in the **training phase** the virtual human is then able to reproduce the motions in interactive training sessions with apprentice users learning the training subject.

The system described in this chapter targets the situation where, in the modeling phase, experts in the training subject are able to model the needed actions and gestures by direct demonstration, without the need of having previous experience with the system. In the training phase, the stored example motions are then re-used

by the virtual human to train apprentice users. More in detail, the system enables the virtual human to reproduce motions with respect to arbitrary target locations in the environment. Figure 4.1 presents one typical scenario modeled using the system.

(a) (b)

Figure 4.1: (a) In this scenario, during the modeling phase, the user demonstrates several examples of pointing motions in order to demonstrate operations with a stereo system and a telephone. (b) During the training phase, the user requests the virtual human to precisely perform the same pointing motions for arbitrary targets, here specified by the apex of the yellow cone which is controlled via a Wii-Mote controller. The training phase is used here to test if the example motions are sufficiently covering the volume of interest in the scenario. The user can interactively switch between the two phases until all required motions are correctly defined. Note that the simulated images in this figure appear fuzzy since they are being projected for stereo visualization.

The motion-based interactive framework allows the design of new types of interaction techniques, which can be developed according to the training scenario at hand. For example, the apprentice may request the virtual human to perform actions at different locations and under different conditions, feedback can be provided based on on-line comparisons between the motions from the expert and the apprentice, etc. Such scenarios are clearly applicable to a variety of training applications, for example, sports training, rehabilitation of motor-impaired patients, training of medical procedures, demonstration of generic procedures, delivery of instructions, etc.

The work presented in this chapter addresses the main computational challenges involved in building such interactive systems. The proposed framework is based on three main computational modules:

- First, a real-time motion capture interface is developed for allowing users to

interactively model motions by direct demonstration. In order to be effective for a variety of situations, our solution includes calibration and mapping from a reduced set of tracking sensors.

- During the motion modeling phase, motions can be recorded and re-played on-line, allowing users to effectively model generic actions and gestures in an intuitive way. The modeling interface also allows the definition of clusters of example motions, in order to represent spatial variations of a same motion. These variations are used during the training phase to precisely parameterize the motions with respect to arbitrarily given spatial targets.

- Finally, given a cluster of example motions built by the user during the modeling phase, an optimization technique for computing motions on-line precisely respecting arbitrarily given spatial constraints is adopted. The technique is based on the *inverse blending* approach. The solution motions are obtained by blending operations with the example motions in a given cluster, therefore the solutions remain human-like and similar to the example motions. This technique is critical for achieving precise parameterizations of realistic actions and gestures, without the need of any pre–computation. Examples are presented with the modeling of pointing and pouring motions, which are precisely parameterized with respect to arbitrarily given target locations.

The interactive motion modeling framework has been implemented in an immersive multi-tile *power wall* stereo visualization system (shown in Figure 4.1). The ability to perform simulations in a large visualization system is important for achieving immersive full-scale interactions, in analogy to the way humans naturally interact with each other. As a result our obtained system represents a powerful and intuitive approach for programming generic parameterized motions by demonstration.

The chapter is organized as follows: after discussing strictly related work in the next section, we present the motion capture interface in Section 4.2 and the modeling interface in Section 4.3. Following, the inverse blending optimization technique is presented in Section 4.4. Finally, results and conclusions are discussed at the end of the chapter.

4.1 Related Work

The approach of direct demonstration of motions, proposed in this dissertation, is strongly related to several imitation-based learning methods previously applied for

different applications in robotics [304, 37] and computer graphics [71, 83]. The work of Cooper et al. [71] in particular also employs a full-body motion capture interface for building a database of motions, however with the focus on building motion databases with good coverage for motion controllers, and not on achieving an immersive and interactive system to teach motions to virtual humans. Similarly, Ijspeert et al. [140] proposed a non immersive system to transfer motions from a human to a robot through learned trajectories. The generated motions are not guaranteed to maintain and to reproduce specific aspects and the quality of the intended motion. Moreover, it relies on a cumbersome full-body motion capture device and it does not take into account motions with respect to virtual environments.

Several methods have been proposed in the literature for addressing the problem of motion reconstruction from a reduced marker set. A popular approach is to employ statistical models [392] and machine learning [59] for extracting from a motion database the motion closest to the input signal. The performance of these methods however greatly depends on the used databases and they are not suitable for real-time applications. Algorithmic approaches based on simulation or optimization have also been proposed [383, 153] but are computationally expensive and are not suitable for achieving human-like motions.

Algorithmic approaches based on Inverse Kinematics (IK) can run in real-time and may be suitable for motion reconstruction if enough markers are provided to limit the overall posture space of possible solutions. Inverse Kinematics has also been employed to optimize full–body postures for tracking the input stream from a reduced marker set [264, 280]. However the convergence time for iterative Jacobian-based solvers over the full–body of a character may require several iterations and can introduce lag in a real-time interface. In fact, the approach proposed by [363] even though very versatile when the training database covers the user's motions, suffers from a severe lag during the reconstruction (0.4 0.6 sec.).

Our interactive interface focuses on achieving a fast solution for reconstructing human-like motions by employing an analytical IK solver [154] applied only to the arms of the character. We then rely on simple mappings from additional markers in order to fully reconstruct upper-body motions very efficiently in real-time. Our present work focuses on modeling upper-body actions but the system can be easily extended to support lower-body reconstruction.

One of the main purposes of the developed system is to model actions and gestures to be used in training applications. Previous work on gesture synthesis has mainly focused on sequencing pre-defined animations [331, 96, 348], or by algorithmic synthesis, such as by employing IK solvers towards specified trajectories [249, 169]. By

modeling gestures with motion blending techniques the system is able to achieve the benefits of both approaches, i.e., realistic animations which can also be parameterized with spatial targets.

As previously mentioned in the general review (see chapter 2.2) the topic of character animation based on motion capture has been extensively studied in the literature for several applications [15, 170, 298, 171, 191]. Although the majority of works focus on the locomotion problem, motion blending (or motion interpolation) has also been well addressed in previous works for modeling gestures and actions.

A specific technique used for addressing spatial constraints is to generate and add pseudo motion examples [170, 296], which however increases the needed computation and storage. The scaled Gaussian process latent variable model [110] optimizes interpolation kernels specifically for maintaining constraints described by latent spaces.

The main limitation of these methods is that alone they are not able to precisely meet given spatial constraints. For instance, the active learning methodology [71] relies on Inverse Kinematics solvers in addition to blending, however risking to penalize the obtained realism.

The proposed method for motion parameterization is based on the optimization of blending weights until best meeting generic spatial constraints defining the target parameterization. The Inverse Blending method can be seen as a post-processing step for optimizing a given set of blending weights, which can be initially computed by any motion blending technique. Only error metrics for the spatial constraints to enforce are necessary in order to optimize the blending weights using a given motion interpolation scheme. The method is able to well parameterize pointing and pouring actions on-line and without the need of any pre–computation.

4.2 Motion Capture Interface

A variety of commercially available solutions are able to map full–body motions to a virtual character in real-time, however the available options are usually expensive and often require the user to wear cumbersome tracking devices.

For instance, retro-reflective motion capture systems require the user to wear a full–body suit with a number of markers carefully placed; systems based on magnetic sensors rely on 15 to 20 magnetic sensors connected with cables; and exo-skeleton systems are heavy and often restrictive.

The main choices that drove the design of our motion capture interface can be summarized as follow: the ability to reconstruct the user's upper body motions with

the minimal introduction of tracking devices; simple, easy to wear and to calibrate; and, the ability to reuse and take advantage of tracking devices already present in a user perspective stereo graphics system (e.g. the user's interaction pointer and the head tracker). Considering the initial requirements, the head tracker and the interaction pointer already produces a precise estimate of the user's hand and head position with respect to the visualization system. In order to estimate the orientation and positioning of the user's body with respect to the scene as well as deriving the user's head orientation and position with respect to the user's lower body (waist/hips) an extra tracker is needed. Finally, to estimate non symmetrical arms motions another tracker needs to be included. As shown in Figure 4.2 (a), our real-time upper-body motion capture solution is based on tracking four key limbs of the user: the two hands, the head, and the lower or mid joint of the spine.

We track both the position and orientation of each of these parts in global coordinates. The user wears simple straps with markers on each of the considered body parts and we rely on a 10-camera Vicon system for performing the real-time tracking. Although we rely on an instrumented room with cameras, our solution can be ported to any other system able to track these four parts. We also rely on a data glove for capturing finger motions in real-time.

(a) (b)

Figure 4.2: The figure shows (a) the camera-based motion interface and (b) our *gesture vest* interface based on inertial sensors [131].

Before starting an interactive session, a calibration process is necessary in order to map the user's body to the skeleton of the virtual human in the scene. We choose not to adapt the dimensions of the virtual human in order to maintain a consistent database of motions which can be shared by different users.

The calibration consists of measuring scaling factors, and requires the user and the virtual agent to stand in a T–pose posture. Let e_i denote the positions of the hands and the head of the user in global coordinates, $i = \{1, 2, 3\}$. Let p be the global position of a point on the user spine at the same height as the shoulder. This point is computed from the spine and hand markers at T–pose. Similarly, let e_i^v and p^v be the same points but computed with respect to the virtual human skeleton. Scaling factors s_i are then computed with:

$$s_i = \frac{\|e_i^v - p^v\|}{\|e_i - p\|}. \tag{4.1}$$

The scaling factors are obtained during the T-pose calibration and then applied during the modeling phase of the system in order to have the end-effector positions of the user and the virtual human matching. Each time new readings are processed, each scaling factor s_i multiplies the translation component of its corresponding body part being tracked, after transforming it to local coordinates in respect of the root joint.

The following additional operations are then performed on-line in order to complete the motion mapping:

- First, the global position of the virtual human is updated according to the tracked spine location of the user. In our current version, we focus only on capturing upper–body actions and gestures without any locomotion, and so no attention is given for tracking or solving leg motions.

- The spine of the virtual human is bent considering the amount of rotation between the tracked spine orientation and the head orientation. This rotation is subdivided by the number of joints being used to represent the spine of the virtual human, and distributed among these joints, similarly to the approach described by Monheit et al. [231]. The rotation of the head joint is directly mapped from the head markers, which are attached to the polarized glasses used for stereo visualization (see Figure 4.2 (a)).

- The arm posture reconstruction (shoulder, elbow and hand) is performed using a 7 degrees of freedom fast analytical IK solver considering arm-body collision avoidance by automatic variation of the arm swivel angle [154]. The solver is applied considering the placement of the shoulder base joint fixed in respect of the spine parent. The swivel angle is set to start at a low default value such that the elbows remain low, as is the case in usual arm motions. In some cases

when the arm is extended, due to tracking and calibration imprecisions, the IK may report that an end-effector cannot reach its target position, in which case we take the closest possible solution. The motion of the fingers is directly mapped from a data-glove. As shown in Figure 4.2 (a), four markers attached at the extremities of two crossed sticks (fixed on the data-glove) are used for tracking the 6 degrees of freedom hand targets.

The mapping produced by this algorithm is extremely fast and results in very fluid interactions always running well above 100 frames per second. Due to the reconstruction of the many user's body degrees of freedom from a minimal tracking subset of objects the algorithm proposed creates plausible poses that are able to capture the characteristics of the motion patterns generated by the user but a precise reconstruction of the overall body cannot be guaranteed.

The algorithm is designed to always grant the correct placement of the character in the scene considering the placement of the user in the working area as well as the character tracked limbs orientations with respect to the user's limbs orientations (i.e. for the head and hands). The positioning in space of the character's limbs in respect of the character's hips are granted to reflect correctly the placement of the user's hands and head in respect of the user's hips as long as the user's pose does not introduce spine bending.

The main limitations of the current approach, leading to inaccurate or erroneous poses, are particularly frequent in the cases explained below. When the user pose introduces spine bending, since we rely on reconstructing the spine joints orientations through uniformly distributing a bending angle, the shoulder placement of character may differ from the user's shoulders placement leading the IK solver to fail reaching the target position. In order to maintain continuity in the reconstruction, as previously mentioned, the closest reachable point to the target position is selected producing straight arm poses that can differ completely from the actual user arm pose. Another erroneous case occurs when the user's hands are very close to the shoulders. In this case the IK solver generates sharp elbow angles and since the arm swivel angle always favor lower values the produced arm poses tend to be maintained parallel to the body. Finally, another failure case is recognizable when the user bend its body maintaining a straight spine.

In order to improve these failure cases and the reconstruction similarity with the user pose the algorithm could be modified to include more tracking devices. For example, to correctly infer the arm swivel angle and remove any reconstruction ambiguity (completely removing the need of using the IK solver) two extra trackers, placed at the shoulder and forearm level can be added to enforce exact fixed rigid

configurations for the arms. This approach is similar to the solution proposed and discussed by the method adopted in conventional motion magnetic capture solutions [293, 292, 350, 347].

Following this approach, we have in particular experimented with an alternate motion capture solution based on our portable and easy–to–wear *gesture vest* prototype system [131].

This device uses five InnaLabs AHRS sensors [142] to capture the orientation changes of performers spine, head and a full arm. In addition, the 5DT data glove is used to capture hand shapes, and a Nintendo WiiMote controller is used for providing basic instructions (record, play, delete, etc) during demonstration of new example gesture motions. As illustrated in Figure 4.1-right, the sensors are attached on a detachable sleeve.

Each sensor measures its orientation in global coordinates based on tri-axial gyro, accelerometer and magnetometer. Each measured rotation is represented in quaternion form with respect to the sensor's canonical reference frame. The maximum update rate is 120 Hz in quaternion mode.

Although it is possible to infer and reconstruct whole-body motions with only a limited number of sensors through models trained on full-range motion capture database [226, 323], the focus of the vest is to capture high quality gesture motions with enough sensors covering the joints of interest, just like the commercially available systems [382, 339] but more cost-effective.

The interface produces very accurate motions, is portable and wireless, and is in particular well suited for capturing one-arm gestures. Figure 4.2 (b) illustrates one interactive modeling section using this equipment. This system is also integrated with a data glove in order to capture hand shapes.

The main drawback of this solution is that alone it cannot track information in global coordinates in respect of our visualization system, making it difficult to be used in our applications related to specification of spatial constraints. When the system is integrated with a global tracking device then it becomes perfectly suitable for our applications. Note that in Figure 4.2 (b) the user is wearing a hat being tracked by the camera system in order to provide global positioning information.

Depending on the application, and on the availability of additional trackers, our gesture vest solution represents a suitable alternative for achieving a small, portable and low–cost solution. The system can be in particular effective for large-scale training scenarios with many users, since the sensors scale well and do not suffer from occlusion limitations.

Achieving a fluid interface has showed to be extremely important for the effective use of the system. The proposed use of a reduced marker set allows the accommodation of systems with fewer (or lower–cost) cameras and also allows the system to be ported to other tracking solutions.

4.3 Interactive Motion Modeling Interface

The ability to run our system in integration with a large immersive display is of main importance in our interactive interface. It allows the user to interact with full-scale virtual environments with immersive stereo vision perception, achieving realistic and accurate reproduction of conditions during both the modeling and training phases.

In order to allow full operation of the system by a single user, our current solution for the motion modeling phase focuses on tracking single-arm actions and gestures performed by the right arm. In this way the user only wears a data glove on the right hand, and the left hand holds a WiiMote controller which provides control over all the system functionality, achieving a simple and effective interactive user interface.

By clicking buttons on the WiiMote controller, the user can change the camera view between several modes, can control the recording and replay of motions, initiate the definition of clustered example motions, add or delete motions from each cluster, etc.

The definition of clusters of example motions is an important concept of our system. The definition of a cluster is necessary for specifying each parameterized action or gesture. When the user selects to start a new cluster, every recorded motion becomes associated with the cluster. Motions in a cluster will be blended during the training phase and therefore they have to consistently represent variations of a same type of motion. For instance, a pointing cluster will contain several pointings of the same type but each pointing to a different location in the environment.

One important piece of information to be associated to each motion in a cluster is its parameterization frame. For example, this frame will be the stroke frame of a gesture or the final frame of a one-way pointing motion. The parameterization frame identifies which frame of the motion is to be parameterized with respect to new target locations during the training phase. We currently let the user specify this frame by pressing a button of the WiiMote controller at the right moment with the left hand, while the motion is being demonstrated with the right arm. The frame location can be then adjusted forward and backwards interactively if needed. This solution is acceptable in several cases but we recognize it may divert the attention

of the user from well performing the motion being demonstrated. We therefore also let the user to interactively select this frame after the motion is performed. We also allow the user to trim the initial and final frames of each recorded example motion.

Clusters of motions can be edited, stored and reloaded as needed. Whenever the user wishes to test a modeled cluster, the system can be switched to training phase and the WiiMote controller is then used to specify targets to be solved by inverse blending. In this way the virtual human is able to perform a new motion precisely reaching given targets and well preserving the quality of the original demonstrated motions. The WiiMote controller also has a haptic feedback which is used during the training phase to tell the user when asked targets are colliding with the environment.

The interconnection of the several modules of our system is further illustrated in Figure 4.3. In the next section we describe our inverse blending optimization technique used during the training phase.

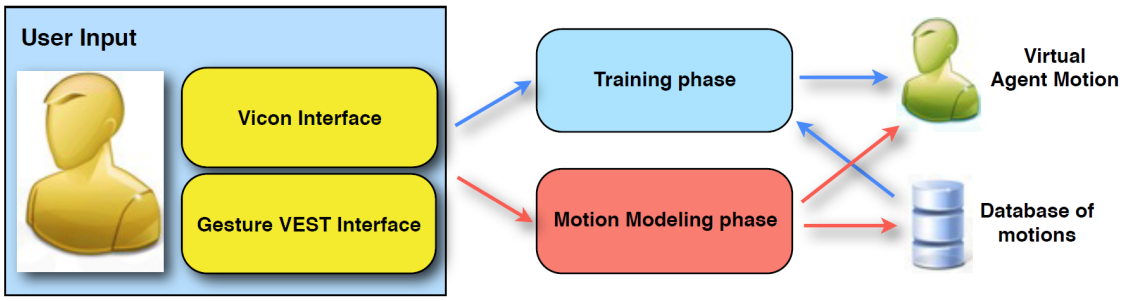


Figure 4.3: Overview of the main parts of the system. The arrows illustrate the data flow during the modeling and training phases.

4.4 Inverse Blending

The optimization core of the motion modeling approach, as previously introduced, relies on the Inverse Blending technique proposed by Huang et al. [132]. For completeness we describe the approach in this section.

The first step for applying the inverse blending optimization is to model each spatial constraint of interest with an error metric function f, which measures how well each constraint is being satisfied at the given parameterization frame. Although the examples presented in this work only use positional constraints for end-effectors, generic types of spatial constraints C can also be taken into account.

Constraints can also have an arbitrary number of degrees-of-freedom (DOF), for example, pointing to a distant location imposes a 2–DOF positional constraint enforcing that the pointing line through the finger reaches a desired target, while precisely pin-pointing a button on a dial pad needs a 3–DOF positional constraint (the target for pointing), and an optional rotational constraint for determining a preferred pointing orientation style (see Figure 4.4).

The optimization starts by selecting k example motions M_j from the example motion cluster that best satisfy the constraint function f, $j = \{1, \ldots, k\}$. For example, in a typical reaching task, the k motion examples having the hand joint closest to the target will be selected. For the case of reaching motions, the hand location at the final pose of the motion is typically used as the parameterization frame. For gestures, the frame of the gesture stroke point is used.

The optimization procedure is based on a traditional but efficient motion blending scheme, where an initial blended motion $\boldsymbol{M}$ is obtained with $\boldsymbol{M}(\boldsymbol{w}) = \sum_{j=1}^{k} w_j M_j$, where $\boldsymbol{w} = \{w_1, \ldots, w_k\}$ are blending weights initialized from a traditional RBF interpolation scheme. Any suitable kernel function can be used and we employ the popular $exp^{-\|\boldsymbol{e}\|^2/\sigma^2}$ kernel. Since our optimization runs on-line during interaction with the user, we do not attempt to optimize kernel functions with respect to the constraints [295, 234]. Instead, the blending weights are optimized independently of the interpolation kernel.

In order to enforce a given constraint C, the goal is to find the optimal set of blending weights $\boldsymbol{w}$, which produces the minimum error e^*, measured by the constraint error function f:

$$e^* = min_{w_j \in [0,1]} \quad f\left(\sum_{j=1}^{k} w_j M_j\right). \tag{4.2}$$

This formulation can also account for multiple constraints by combining the error metric of each constraint in a single weighted summation. Two coefficients are then introduced for each constraint C_i, $i = \{1, \ldots, n\}$: a normalization coefficient n_i and a prioritization coefficient c_i. The purpose of coefficient n_i is to balance the magnitude of the different error metrics associated to each constraint. Coefficient c_i allows the specification of relative priorities between the constraints.

The result is essentially a multi-objective optimization problem, with the goal being to minimize an error metric composed of the weighted summation of the individual error metrics:

$$e = \ min_{w_j \in [0,1]} \sum_{i=1}^{n} \left(c_i \ n_i \ f_i \left(\boldsymbol{M}(\boldsymbol{w}) \right) \right). \tag{4.3}$$

Independent of the number of constraints being addressed, when constraints are fully satisfied, $e \to 0$. Figure 4.4 shows several results obtained by the optimization scheme.

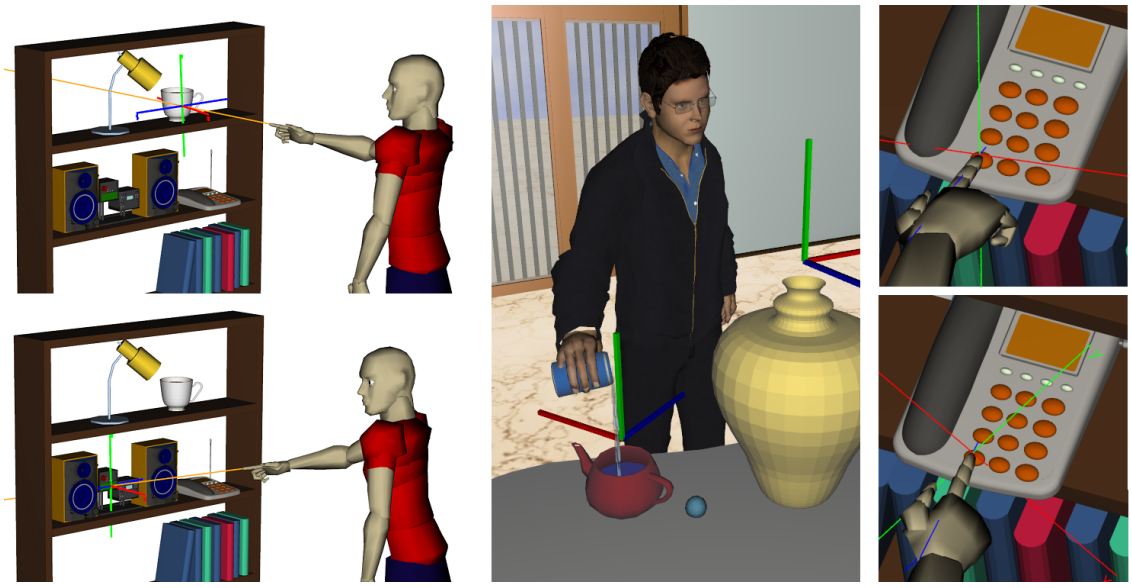


Figure 4.4: The image shows results obtained with three different motion clusters. (a) Pointing motions parameterized by a 2–DOF directional constraint results in precise pointing to distant targets. (b) Pouring motions can be parameterized by a 2–DOF planar constraint specifying the precise location above the pouring target, and an additional constraint specifying an acceptable height range, so that liquids can correctly flow down into containers. (c) Precise pinpointing to given targets requires a 3–DOF positional constraint, with optional rotational constraints for further controlling the final poses obtained. The shown pinpointing examples show different orientations obtained, which match the x-axis of the tri-axes manipulator.

Several optimization routines were implemented for solving the inverse blending problems, including: steepest ascent hill-climbing, the Nelder-Mead method and the gradient descent method [274]. Performance evaluations were conducted by solving 5000 inverse blending problems for different scenarios: pointing, pouring and grasping. The Nelder-Mead method [274] has been proved to be the method of choice

for our case where k remains below 15. The method requires a simple implementation and can typically achieve optimal blending weights within 2 milliseconds of computation time.

With suitable example motions in a given cluster, inverse blending can produce motions exactly satisfying given spatial constraints and fast enough for real-time applications. The several examples presented by this approach demonstrate its successful execution in different scenarios. To evaluate the performance of our method, a reaching task was designed to measure the errors produced by our method against a single RBF interpolation, with the 16 reaching motions in the database from Mukai and Kuriyama [234]. A total of 114 reaching goals (each specifying a 3–DOF positional constraint) were placed evenly on a spherical surface within reach of the character. These goals are highlighted with small yellow dots in Figure 4.5. The end locations of the hand trajectory in each example motion are shown as gray dots.

For each reaching target on the surfaces shown in Figure 4.5, the RBF interpolation alone is used first to generate a reaching motion and record the final hand position where the character actually reaches. These final positions are used to construct a mesh grid, which is shown on the upper row of Figure 4.5. Each triangle on the mesh is colored with respect to the average errors from its vertices, representing the distance error between the final hand positions and their corresponding reaching targets. We then use inverse blending optimization to perform the same tasks, and the mesh constructed is shown on the lower row of Figure 4.5. The reaching motions generated by inverse blending can precisely reach most of the targets, and the measured errors were practically zero across most of the mesh. Only at the boundary of the surface the errors start to appear. In this specific task, the radius of the spherical surface was set to $80cm$, and both methods used eight example motions from the database ($k = 8$) for computing each reaching task.

It is important to note that the ability of enforcing constraints greatly depends on the existing variations among the used motion examples being blended. The number of needed example motions also depend on the size of the target volume space. The computational time required for finding solutions will also depend on the quality and number of considered motion examples (the k value). However, as showed in our examples, these limitations can be easily addressed by appropriately modeling example motions, and balancing the coverage vs. efficiency trade-off specifically for each action being modeled.

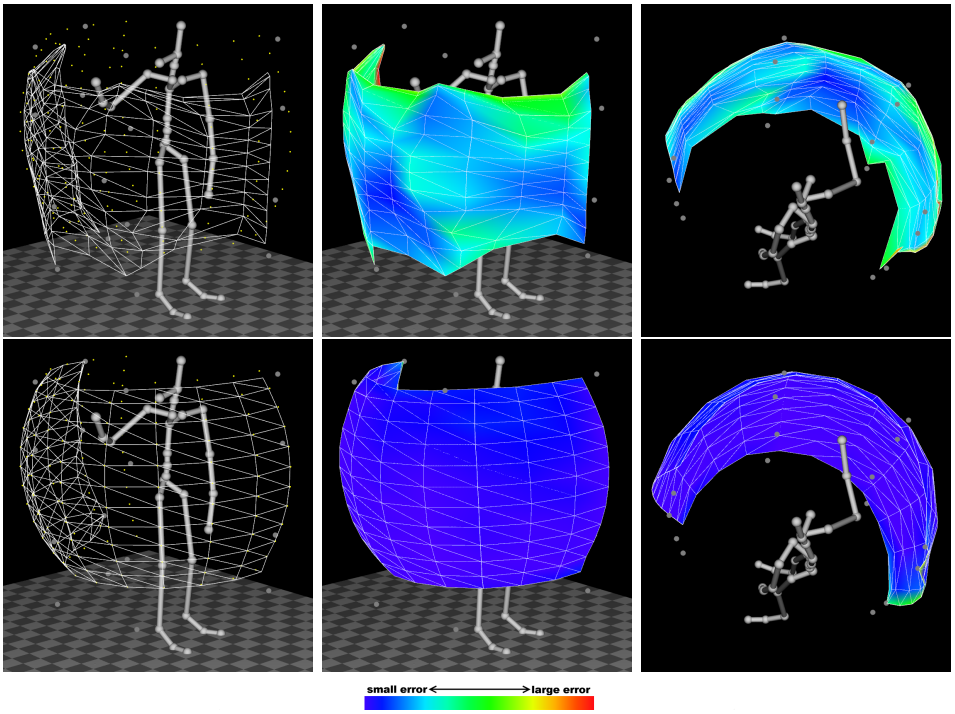


Figure 4.5: Visualization of errors obtained by RBF interpolation and inverse blending. The upper row shows the results obtained with RBF interpolation. The blue smooth meshes on the lower row show the inverse blending results, which can easily satisfy the given 3–DOF positional constraints.

4.5 Database Coverage Visualization and Refinement

Two specific visualization methods, following the ergonomic design approach of using a palette of colors inside the workspace as visual guidance [386, 291, 389], have been developed to intuitively guide the user during the process of adding new motions to refine the database for improved coverage. The visualization tools developed are an adaptation of the visualization tools proposed by Huang et al. [130] to be used in the interactive motion modeling framework in real-time.

The presence of these visualization tools in the motion database creation process is a key factor since the existing variations among the example motions being interpolated greatly affects the ability to enforce constraints using inverse blending. It is important to determine which example motions to capture to ensure that a well-built cluster of motions is formed, with good coverage of the regions of interest

(ROIs) inside the workspace. To the contrary, defining an overly fine subdivision of the constraint space with too many examples is inefficient and impractical as it requires capturing too many example motions to populate a database. Instead, since similar examples can often be interpolated to produce valid new motions with good quality, a small number of carefully selected example motions is better in providing good coverage for the ROIs in the workspace. Achieving an efficient database is also key to ensure interactivity of the system.

The two visualization methods proposed consist on a global Workspace Volume Visualization (WV) and a Local Coverage Visualization (LV) to query specific region of the space interactively.

4.5.1 Workspace Volume Visualization

WV conducts a coarse uniform sampling of the workspace and presents the overall spatial coverage with colored cubes for the entire workspace without the need to define an overly fine subdivision of the constraint space. Each cube represents a reaching target (spatial constraint), and a motion synthesized towards each cube is measured by reaching precision (error e^*) using a constraint evaluation function, and the value $e^* = e_{max} (\in [0; 1])$ is mapped onto a hue color space then assigned to each cube. For a reasonably sized database WV takes a few seconds to generate, then the user can immediately spot areas with low coverage by the color of the cubes (red or orange), and add additional motion towards these areas.

4.5.2 Local Coverage Visualization

In certain cases, the global error-based volume visualization is not needed when the user is fine tuning the coverage of a small region, or when only a small local region is of interest. In addition, the pre-computation time can impose undesirable lags when editing large motion sets. These can be solved with LV.

LV renders a transparent colored mesh geometry covering a small ROI, delimiting the coverage evaluation within its volume. It focuses on the local coverage visualization taking only milliseconds to be computed, and it is suitable for fine tuning coverage of smaller volumes when only small local regions are of interest. LV uses the same color mapping as WV, and the error ratio is assigned to the corresponding vertex. Color on the mesh surface comes from Barycentric color interpolation with Gouraud shading.

LV follows the movement of the user's hand, its size, shape and resolution can be

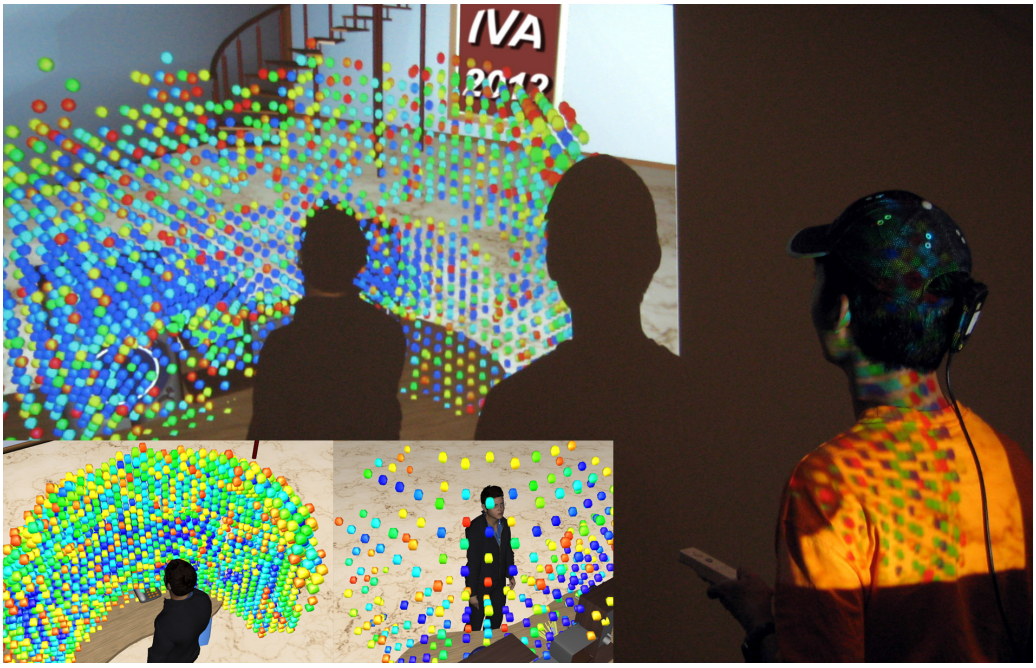


Figure 4.6: Workspace Volume Visualization mode gives an overview of database coverage. The sampling density and error threshold can be adjusted for clear viewing.

iteratively changed with the WiiMote controller for either fast sweeping over large ROIs (a table surface) or for carefully checking small ROIs (buttons, etc). LV is also able to utilize motions dynamically added to the database in real-time applications without any pre-computation lag. Details on motion synthesis and error evaluation with spatial constraints can be found in [132].

4.6 Conclusions

The system developed achieves an effective overall design for modeling parameterized motions, which are extremely important for several types of applications (see Figure 4.8). The framework addresses the modeling of generic gestures and actions to be executed by interactive virtual humans, and furthermore allows non–skilled animators to intuitively obtain realistic results.

This chapter presented a novel motion modeling framework based on the direct demonstration and parameterization of motions. In particular the chapter presented several algorithmic solutions required for enabling the development of the proposed

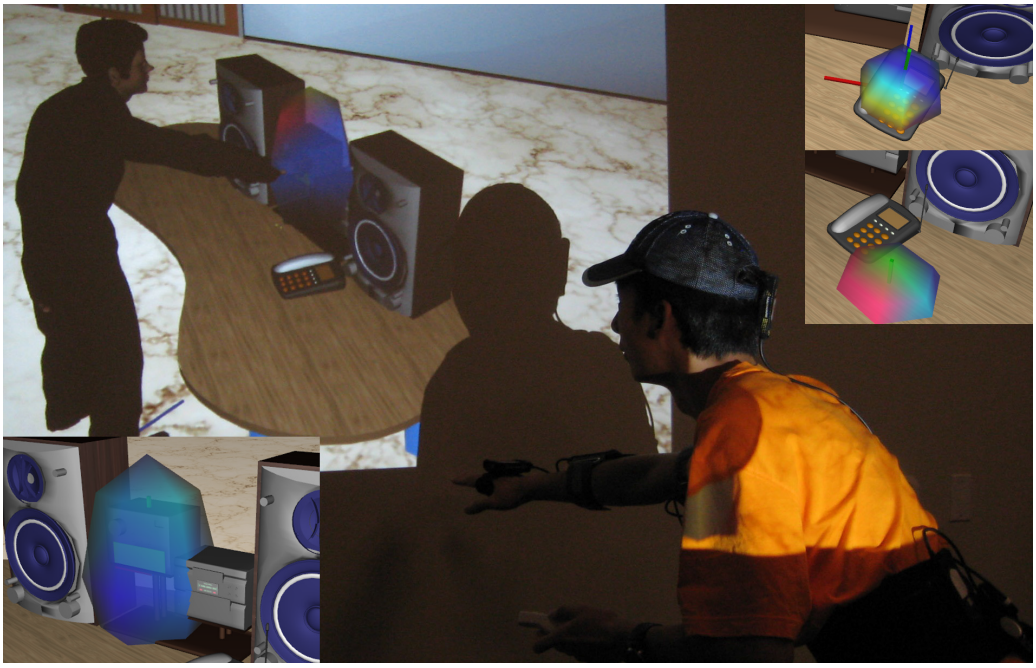


Figure 4.7: Local Coverage Visualization mode. The rendered surface follows the movement of the user's hand, ideal for checking small ROIs.

design: a fast procedure for motion mapping from a reduced marker set, an intuitive motion interface for enabling the direct demonstration of parameterized actions and gestures, and an inverse blending optimization technique able to efficiently achieve realistic and parameterized motions in respect of arbitrarily given targets.

The coverage visualization tools effectively allow the user to observe the database coverage immersively within the workspace and during the on-line collection of example motions. The interactive visualization is able to guide the user to concentrate on capturing motions where they are needed, in regions with less coverage. The user can thus intuitively construct suitable databases for parameterized actions or gestures with guaranteed coverage within the specified precision. The platform could be adapted for different hardware configurations.

The proposed framework has been implemented in an immersive multi-tile stereo visualization system, achieving a powerful and intuitive interface for programming generic parameterized motions by demonstration.

The overall concept of the system has the potential to impact many applications. Our first results open several new opportunities for further development of our inter-

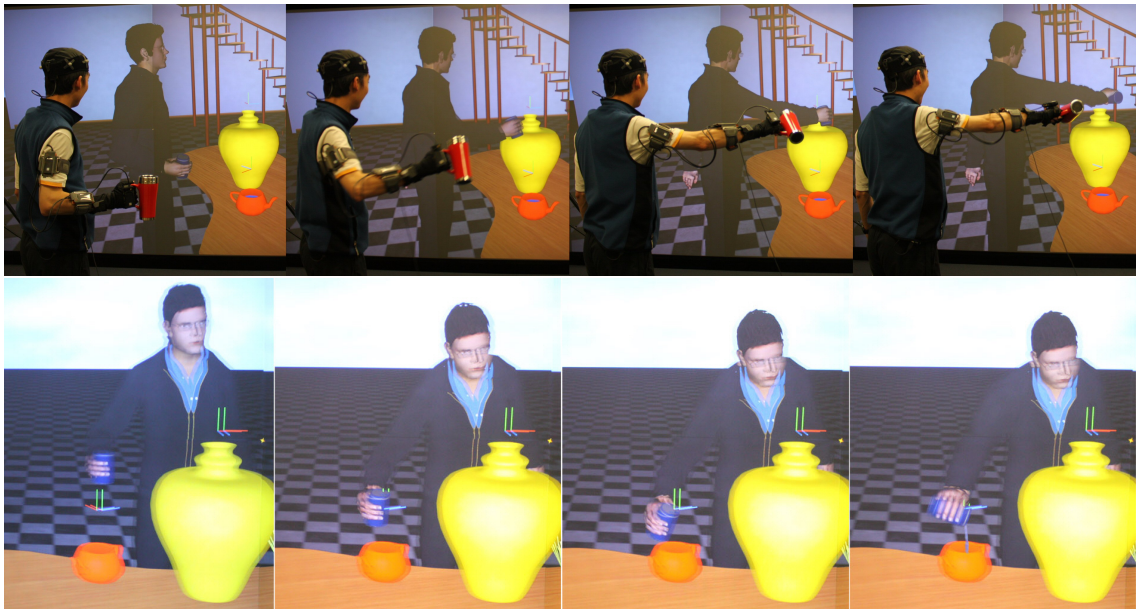


Figure 4.8: In this example pouring motions are demonstrated and added to a parameterized pouring action cluster (top sequence). Once completed, the cluster is used to generate new pouring actions to arbitrary locations (bottom sequence).

face design but still several improvements can also be performed. For instance, one drawback of the parameterization solution adopted is that the ability of enforcing constraints greatly depends on the existing variations among the motion examples being blended and also the computational time required for finding solutions will also depend on the quality and number of motion examples. However, as shown in the system, these limitations are easy to address by appropriately modeling example motions, and balancing the coverage versus efficiency trade-off.

Considering future improvements, a two-arms and hands tracking interface can be easily integrated by including a few voice commands and/or finger and gestures recognition. The integration of locomotion would be also beneficial to achieve the scope of creating a generic training system. Automatic locomotion and body positioning algorithms for controlling the virtual human are necessary for creating effective and believable virtual tutors in virtual environments. Moreover, an independent gaze model, as proposed by [133], also appears to be necessary for creating believable virtual tutors.

CHAPTER 5

Perception and Task Evaluation in Varied Immersive Systems Configurations

Humans are highly social and possess exceptional skills for communication with other humans. A natural approach for immersive virtual reality systems is thus to rely on interactions that are as close as possible to how humans interact with each other. Animated characters and avatars often emerge as key elements for replicating human forms of communication, improving usability and accessibility to all types of users. The approach is promising for several applications in education, training and rehabilitation [149, 381].

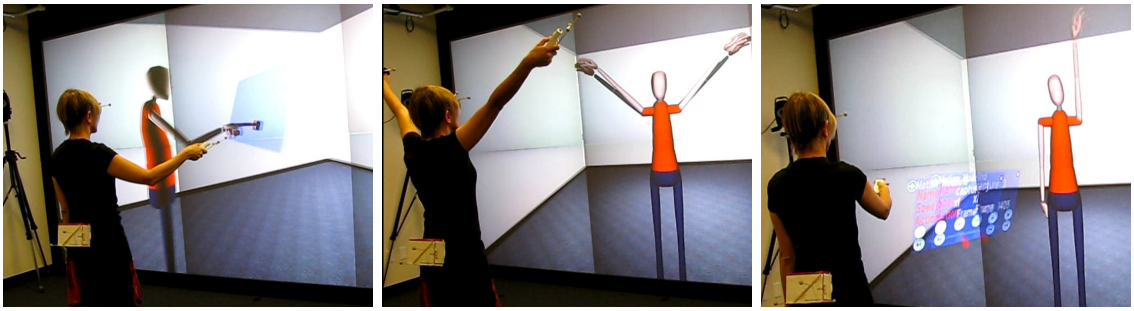


Figure 5.1: Illustrative snapshots of the experiments: target reaching (left), motion reproduction (center), and motion modeling (right).

As previously mentioned in Chapter 4, a powerful approach to achieve natural user interactions is based on the concept of motion modeling by demonstration where, first, an expert human instructor demonstrates to an autonomous character how tasks should be performed, such that later the autonomous character can deliver the training material autonomously to several users. A common scenario proposed by this approach can be found in motion-oriented training and rehabilitation [360, 11]. In these situations, a virtual coach or tutor can naturally demonstrate and monitor user performances, collecting important data for post-analysis.

In this chapter we investigate the effect of different system configurations on user

performance. Because numerous variations are possible, this research focuses on specific configurations in three experiments that are particularly relevant to rehabilitation and training applications. Minimal user instrumentation is important in rehabilitation, thus we have not included experiments with head-mounted displays.

The first experiment focuses on reaching tasks (Figure 5.1-left). Reaching represents an important class of motions used in exercises for rehabilitation of fine arm motor control [349]. The experiment was designed to analyze the effect of different configurations on the reaching tasks. The chosen configurations reflect typical choices that are made in practice, and the obtained results provide new observations and quantified information on the trade-offs between the varied conditions. For example, our results are important to guide the choice of adopting an expensive and space-consuming display configuration instead of a common desktop setup.

The second experiment was designed to analyze the ability to perceive and subsequently reproduce motions demonstrated by an autonomous virtual character (Figure 5.1-center). This scenario is important because motion reproduction is a key activity in several exercise and therapy applications. The motions of several subjects were captured and compared under different conditions, providing new information on the effects of using avatars and stereo vision.

The third experiment is a usability study that analyzes user experiences in a complete immersive interface to model motions by demonstration (Figure 5.1-right). This study was selected in order to give insight in the usability of the approach in real applications, for the design of rehabilitation exercises or task-oriented motions for generic training scenarios.

The selected tasks and configurations expose new important trade-offs between different forms of direct task execution and avatar-based visual feedback.

5.1 Related Work

This chapter evaluates specific scenarios with attention on the different ways of employing animated characters and avatars. The addressed factors have only been studied before in isolation, and did not investigate the impact of virtual characters and avatars on task execution. In the following related work analysis we first review perception studies related to immersive systems, and then studies related to virtual characters.

5.1.1 Evaluation of Immersive Systems

The effects of immersive virtual reality on scientific visualization, data analysis and on human interaction have been previously studied in different ways. Depth perception through stereoscopy has been demonstrated to reduce time and error, and to improve user performance in spatial tasks [365, 277]. A frequent problem in any type of virtual environment is distance misestimation [351], which has been detected in both real workspace measurements and egocentric distances. The reason behind this behavior is not clear, and it has also been detected in head mounted displays (HMDs) and in stereoscopic wide screen displays (WSDs) [371, 374, 167]. Interestingly, Naceri et al. [236] have found distance underestimation to be higher in HMDs than in WSDs.

Display size has also been investigated, and large displays have been reported to be beneficial in spatial tasks [341]. In particular, Ball et al. [25] studied the effectiveness of large high-resolution displays for interactive data visualization, concluding that a large display is preferable because it minimizes the use of virtual tools for navigation control. Considering a display physical field of view (PFOV), it has been shown that a wider PFOV can yield significantly better performance than a smaller PFOV in hand-eye coordination tasks [8], and in search and comparison tasks [18, 25, 244]. PFOV has also a direct impact on spatial awareness, memory and presence [199]. Ni et al. [244] have conducted experiments showing that large displays and high resolutions improve user performance in search and comparison tasks.

Some previous studies have also considered multivariate evaluations of combined factors. Stereoscopy and head tracking have been found to have significant impact on spatial understanding [277, 219] but not necessarily on object manipulation tasks [238]. Display and interaction modes have been observed to significantly influence a user strategy and performance in a virtual reality game [218]. The results were in favor of real world settings (high-resolution display with user-perspective interaction) or simple game-like interaction (low resolution display with common mouse/keyboard interaction).

Evaluation studies are important to guide the development of effective rehabilitation applications, which have become particularly popular in a number of cases involving arm motions, such as in post-stroke rehabilitation [86, 47], reachable space measurement [115], etc. We provide in this research new results on the effects of different configurations involving animated characters and avatars on reaching and motion reproduction tasks, which are important tasks in rehabilitation systems.

5.1.2 Animated Characters and Avatars

The use of animated characters as mediators in virtual environments is a natural approach to replicate human forms of interactions. Human-human communication can however be highly complex, involving several multi-modal processes [38]. In particular, psychologists have shown that people are remarkably skilled in recognizing the features of a person through his or her motions; for example, when identifying gender [172], emotions [21], or the identity of a known person from just a synthesized motion silhouette [74].

The subtleties of human perception with respect to virtual characters have been explored in different ways. Visual artifacts and unnatural animations have been observed to lead to negative user reactions [45], and user sensitivity to errors in synthesized human motions have been studied in the context of ballistic motions [282]. Other types of studies have targeted rendering styles [216] and how character appearance influences the perception of actions [60] and bodily emotions [217]. The use of avatars has also been investigated with respect to user embodiment, ownership and behavior [39, 328, 265].

Our work focuses on investigating how different ways of using avatars and animated characters influence the execution of motion-oriented tasks. The presented results expose trade-offs not investigated before, related to task execution in user space versus avatar space and under different conditions.

5.2 Experimental Design

In this session we describe the overall experimental design of the three reported experiments. The experiments are illustrated in Figure 5.1, and they are later described in detail in sections 5.3, 5.4 and 5.5.

5.2.1 Apparatus

The experiments were performed in our virtual reality lab and they were designed to run in a large immersive stereo vision display wall (UC Merced's Powerwall) or in a regular desktop machine.

The Powerwall visualization system is a retro-projected surface of $4.56m$ by $2.25m$ illuminated by twelve projectors (each 1024x768@60Hz) with circular passive polarization filters. The projectors are connected to a rendering cluster of six commod-

Table 5.1: Summary of user groups for each experiment. Letters F and M in the second column specify the number of female and male participants. Experiments are labeled with the explained 3-letter acronyms.

Group	Participants	Experiment	Label	Description of the Corresponding Configuration
1	10 (F: 7 M: 3)	**1**	**SLU**	*Stereo vision. Large display. User-perspective direct interaction without the use of an avatar.*
		2	**SLT**	*Stereo vision. Large display. Tutor (as an autonomous virtual character) used to assist with the interaction.*
2	10 (F: 6 M: 4)	**2**	**MLA**	*Mono vision. Large display. Avatar of the user is displayed during the interaction.*
		1	**SLA**	*Stereo vision. Large display. Avatar of the user is displayed during the interaction.*
3	10 (F: 5 M: 5)	**1**	**MLA**	*Mono vision. Large display. Avatar of the user is displayed during the interaction.*
		3	**SLA**	*Stereo vision. Large display. Avatar of the user is displayed during the motion recording phase.*
4	10 (F: 5 M: 5)	**2**	**SLA**	*Stereo vision. Large display. Avatar of the user is displayed during the interaction.*
		1	**MDA**	*Mono vision. Desktop-based small display. Avatar of the user is displayed during the interaction.*
5	10 (F: 3 M: 7)	**3**	**SLN**	*Stereo vision. Large display. No avatar was used during the motion recording phase.*
		2	**MLT**	*Mono vision. Large display. Tutor (as an autonomous virtual character) used to assist with the interaction.*

ity Linux-based rendering nodes (Pentium Q9550 2.83GHz GeForce GTX 280 4Gb RAM) driven by a similar main machine controlling the virtual scene being displayed. The cluster is connected through a gigabit ethernet.

The virtual reality lab also contains an optical 10-camera Vicon motion capture system that provides sub-millimeter tracking precision. The system was used to track the user's head position (for user-perspective stereo rendering), the interaction device held by the user, and two other set of markers for tracking the free hand and the torso. The upper-body motion of the user was then reconstructed from the tracked information. The interaction device, used primarily for button input, was a Nintendo Wii-mote controller.

The desktop configuration consisted of the main node computer previously described, which was connected to a standard 32 inches display (1920x1080@60Hz), without stereo vision. In each activity the application of the experiment was running in full-screen.

5.2.2 Participants

Fifty participants took part on the experiments. The participants were divided in groups of 10 people randomly generated according to each experiment day and availability. In order to well cover all considered variations, each participant was assigned to perform two different experiments sequentially, with the order of execution rotated every five users. It is possible that the choice of reusing participants may have influenced familiarity and thus the results; however, we believe that this effect has been highly minimized due to the unrelated experiments and the varied execution order. The group assignments and system variations are summarized in Table 5.1.

The participants were undergraduate students selected randomly from a pool of students enrolled in the university's experiment management system (students of Engineering, Natural Sciences, Social Sciences or Humanities disciplines). The demographics varied from 18 to 25 years old and 26 participants were female. Because of hardware and tracking volume limitations few restrictions were imposed during the participant selection: color blind, stereo blind (monocular/flat-vision), motor impaired or taller than $1.85m$. Although the system required the use of the right hand during the reaching tasks, we did not enforce the requirement of having right-handed participants. Four participants were left-handed.

Three questions were used to estimate the familiarity of the participants with the involved technologies. A total of 46 out of the 50 participants considered themselves very familiar with electronic devices (smartphones, computers, tablets, etc.); 36 participants declared to be very good with first person shooting and role-play video games (games where avatars/characters are involved) and 3 knew or had already used an immersive user-perspective stereo vision system before.

5.2.3 Materials

Participants were required to wear or hold four objects with attached markers that were tracked by our optical tracking system: the stereo glasses, the Wii-mote controller (held with the right hand), a bracelet on the left hand, and a belt. These four tracked objects were needed to achieve user-perspective stereo vision with calibrated real-virtual dimensions, and to reconstruct the user's upper-body motions in his or her avatar in real-time.

Before the start of each activity an instruction sheet was handed to the participant. The instructions consisted of text and pictures explaining the application scenario, the controls, and the task to be performed, in bulleted lists (but well de-

tailed) explanations. At the end of each task participants were asked to fill a paper questionnaire with questions related to preferences, usability and user experience. Questions were both open and based on the Likert scale.

5.2.4 Procedure

Each participant session was organized in four phases: informal demographics questionnaire, introduction to the system and training, first activity, and second activity. Activities were performed in four steps: avatar-user calibration, activity learning, execution, and debriefing. The total time taken per user was around one hour, with short breaks allowed.

When ready, the participant was equipped with the trackers and positioned to execute a training scenario with the Powerwall display. The training scenario consisted of a simple user-perspective object manipulation environment that included floating panels with a virtual interaction pointer. The scenario allowed the user to manipulate virtual objects and get used to the system interface. In general participants took from 10 to 15 minutes training. The scenario included a virtual room extending the real laboratory room (same wall color, carpeting, etc.), and it was designed to minimize distractions from the task. This same background scenario was used in all experiments.

Following the training step, the instructions sheet for the next activity was handed to the participant. A summarized bulleted list of the task was also provided to help the participant memorize the task. Each activity involved minimal memorization of procedures, for example: place arms along the body, click button on the controller when ready, raise arm to drive the controller toward a target, click the button when satisfied, repeat when ready, etc. The participant was allowed to take the needed time reading the instructions and preparing for the task. Activities were each completed in about 5 minutes.

Each activity required a simple calibration procedure where the participant would perform a simple T-pose required for mapping his or her dimensions to the avatar, as described in Chapter 4. During the activity, participants were not allowed to step away from their initial placement and to communicate with the researcher. After each activity the participant then completed the follow-up questionnaire.

5.3 Experiment 1: Reaching Targets

The first experiment investigated the accuracy of reaching virtual targets under different configurations. The variations included the avatar use, the screen size, stereo vision and the use of user-perspective direct interaction. Forty participants took part in the experiment and they were divided in group of 10 participants. Table 5.2 summarizes the four variations (with participant's gender balance), and Figure 5.2 illustrates the experiment.

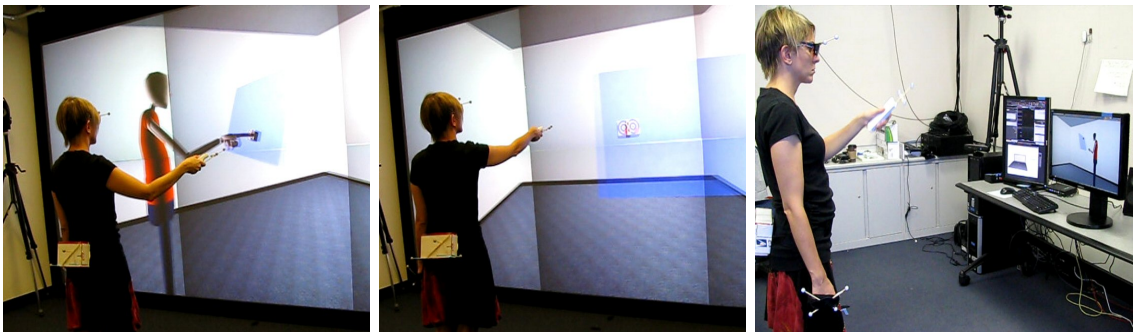


Figure 5.2: Experiment 1 investigated the influence of avatars, stereo vision and display size on user performances during reaching tasks. Variations (see Table 5.2): SLA and MLA (left); SLU (center); and MDA (right).

Among the several combinations possible, only combinations that made sense in practice, and that could be reasonably implemented, were considered. For example, the small monitor configuration was not suitable for stereo vision because users had to perform the tasks standing and at a certain distance, and the stereo effect could be easily lost due the limited field of view during the interactions.

Table 5.2: Configurations of experiment 1 (G = Group, M = Male, F = Female).

Label	G	M	F	Screen	Stereo	Avatar	View
SLU	**1**	3	7	*large*	*yes*	*no*	*first-person*
SLA	**2**	4	6	*large*	*yes*	*yes*	*third-person*
MLA	**3**	5	5	*large*	*no*	*yes*	*third-person*
MDA	**4**	5	5	*small*	*no*	*yes*	*third-person*

Three variations of the experiment (SLA, MLA and MDA) included the user's avatar standing in front of a floating surface. The task consisted of reaching virtual targets spawning on the surface in front of the avatar. The flat semi-transparent surface had a vertical inclination of 20 degrees and the target objects (white cubes

with red concentric circles) appeared on top of it one at a time. The avatar appearance was designed to be simplistic and with little facial detail, in order to drive the user's attention to the avatar's motions and to minimize perceptual distractions due to visualization artifacts or inexpressive gaze or face.

The task made the user to control his or her avatar's right hand index finger tip towards the center of the current target cube, being as accurate as possible. The upper-body motions of the user were directly mapped in real-time to the avatar. The avatar was thus mimicking the user's motions and, since the user's point of view was from behind the avatar, no motion mirroring was implemented.

The user's point of view was from a lateral/anterior position, such that the whole working surface and the avatar's right arm motion was clearly visible at all times. Proximity and full visibility are important because otherwise the user would experience an additional cognitive that could impact task performance.

The three avatar-based variations differed from each other only by the type of visualization. The first group (SLA) worked with a large screen with user perspective stereo vision enabled, the second group (MLA) worked with the large visualization surface without user perspective vision (only simple mono vision), and the third group (MDA) performed the task in front of a desktop display without stereo vision. In this last variation the users were placed at $1.5m$ from the main screen, which was placed at a comfortable height. This setting was designed to emulate a user interacting with an inexpensive tracking device such as Microsoft Kinect or similar in a possible home setup. The distance selected is the optimal distance that would grant to the sensor enough field of view to optimally track the user's body.

Users performing the variation with the avatar and stereo vision (SLA) were able to perceive the scene in a spatially calibrated and metrically correct fashion. In this variation they were placed in front of the screen at a distance that allowed them to perceive the character at the approximate distance of $1m$ away.

The fourth variation, user-perspective direct interaction (SLU), was similar to SLA but with the virtual avatar not being used and instead the user directly interacted with the virtual scene. The working plane and targets were thus perceived by the participants as directly floating in front of them, enabling the participants to directly perform pointing actions toward the targets. The virtual pointer was rendered in the scene always floating at $10cm$ in front of the interaction controller and the user was asked to drive the virtual pointer towards the center of each target.

The task execution started with the participant standing in a comfortable rest position with arms down along the body. A new target would appear by pressing a button in the Wii-mote interaction device. The user was then required to move his

or her right arm until the avatar's index finger would touch the center of the target. The user was asked to pay particular attention on precisely reaching the target's center. When the user was satisfied with the positioning, pressing the interaction button again would complete the task. These steps were then repeated ten times per user. The targets appeared in five different locations, regularly distributed in the surface, but not following any pattern so that the targets were perceived to be randomly placed.

5.3.1 Data Collected

In all variations except SLU, the motions of the avatar were collected and saved for analysis. Motion files were represented as time series of joint angles expressed locally in the hierarchical skeletal representation of the avatar. Given the initial scaling calibration parameters, it was possible to reconstruct both global positions and relative distances to virtual objects such as for measuring the error with respect to reaching the center of the targets. For each motion collected, the global trajectory generated by the fingertip of the right arm of the avatar was extracted for analysis.

For the user-perspective direct interaction variation (SLU) the body motion data were not used and instead the time-stamped global positions and orientations of the virtual pointer were extracted and collected per action performed. The motions of the participants and the virtual pointer trajectory were recorded at 30 frame per second for all performed activities.

Finger and pointer trajectories were segmented and clustered according to the five targets reached. We have noticed that trajectories exhibited two distinctive phases: an approach phase and an adjustment phase. Given a motion M, let t_{ph} denote its phase transition time point. The approach phase of the motion M_{ap} is the initial part of the motion where the user quickly moved his/her arm towards the target, and the adjustment phase M_{ad} is when the user spent time to adjust the end-effector on the target center as accurately as possible. Transition point t_{ph} was manually annotated per motion, considering the first occurrence of a sudden deceleration or change of direction in the trajectory. We have used a rule that analyzes if the group of frames inside 1 cm diameter sphere (usually 10 to 15 frames) show a sudden change in trajectory direction, of more than 45 degrees. In very few cases in SLU this point was not observed and the t_{ph} value received the full duration of the motion.

In order to investigate the performances we have considered several descriptive parameters such as trajectory durations, phase transition times, average velocities and accelerations, and distances to the target. Distances to the target were measured

from the end-effector position (character's fingertip or virtual cursor) to the target center. An extract of the descriptive statistic analysis is shown in Table 5.3.

Table 5.3: Experiment 1 descriptive statistics extract. Notation: t_e: overall trajectory duration, d_e: distance to the target at t_e, v_{avg}: average velocity, a_{avg}: average acceleration, t_{ph}: time of phase change, t_{phr}: time of phase change relative to t_e, d_{ph}: distance to the target at t_{ph}, d_σ: standard deviation of the target distance during M_{ad}. Except for the last row, the shown values are mean values with the standard deviation in parenthesis.

p.	unit	**SLA**	**MDA**	**MLA**	**SLU**
t_e	s	5.803 (2.06)	6.943 (1.34)	5.963 (1.24)	2.526 (1.28)
d_e	m	0.029 (0.04)	0.080 (0.31)	0.100 (0.04)	0.005 (.004)
v_{avg}	m/s	0.282 (0.08)	0.249 (0.02)	0.271 (0.04)	0.657 (0.12)
a_{avg}	m/s^2	3.420 (2.75)	2.287 (1.59)	3.043 (2.74)	19.03 (7.24)
t_{ph}	s	1.923 (0.69)	2.333 (0.46)	2.229 (0.37)	1.414 (0.36)
t_{phr}	%	38.22 (10.1)	38.88 (10.9)	41.73 (8.00)	64.24 (14.2)
d_{ph}	m	0.100 (0.06)	0.124 (0.05)	0.162 (0.03)	0.024 (0.01)
d_σ	m	0.064	0.085	0.074	0.016

We have also considered trajectory profiles on distance to the target, velocity, and acceleration at every frame. In order to generate comparable data we have uniformly time-warped and normalized the profile samples. The readings were grouped by participant repetition and by target with the purpose of analyzing if the target location would affect performances. Since the targets were placed in comfortable reaching positions we did not observe any significant difference related to target placement.

5.3.2 Results

From our initial analysis we have clearly found, as expected, that users were faster and more accurate when using the direct interaction configuration using user-perspective stereo vision (SLU). Figure 5.3 shows the differences in the average time, phase transition time and final distance to the target. In SLU users were almost six times more accurate than the best solution using avatars SLA (on the range of $5 \pm 4mm$ against $3 \pm 0.4cm$ in SLA) and they were twice as fast (around $2.5 \pm 1.3s$ against $5.5 \pm 2.0s$ average time). Considering the difference in phase transition time (t_{ph}) SLU participants were 5 times closer to the target's center already at point t_{ph}, in comparison to SLA, which is the best avatar-based configuration. At the end of the task they were 6 times closer to the targets than in SLA. This fact explains the main

difference in velocity and the interval velocity variance during the reaching phase with a subsequently shorter adjustment period. Figure 5.4 shows the normalized velocity profiles resulting from each trajectory clustered and averaged by variation where, clearly, SLU is faster than the other three methods and t_{ph} occurs closer to the t_e.

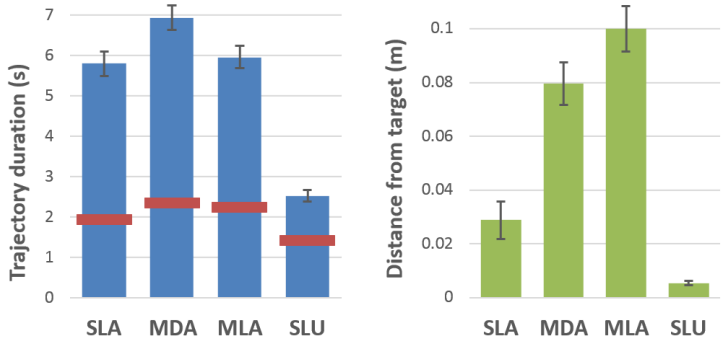


Figure 5.3: Left: each bar represents the average trajectory duration t_e for each variation in experiment 1. The horizontal segment mark depicts t_{ph} in relation to the average t_e. Right: each bar represents the average target distance d_e. In both graphs, the vertical line ranges show the standard error.

Considering the three variations with avatar interaction, Table 5.3 reveals that the configuration with stereo vision (SLA) offers the best results. Although the overall reaching time and velocity is similar and the adjustment phase can be comparable (only slightly better in SLA), SLA participants generated motions that are three times more accurate than the other two variations. Moreover, the reaching phase during SLA led to a shorter adjustment phase where the overall standard deviation distance was lower. Surprisingly, it is possible to observe that the two variations with 2D vision are very similar in terms of accuracy, with the accuracy achieved in MLA being slightly lower than in MDA. But the group using the small desktop display (MDA: $0.249 \pm 0.02 m/s$) performed each task slower than the group interacting with the large display (MLA: $0.271 \pm 0.04 m/s$).

Considering the data gathered from the post-experiment questionnaire (expressed in a Likert scale from 1 to 7), we noticed that participants belonging to MDA expressed a lower level of confidence, in terms of accuracy and awareness of being precise, in comparison to the participants using the large display (the system accuracy perceived in MDA was 3.6 while in the other three variation was higher that 5.0). Most of the participants believed that the limitation was given by the setup

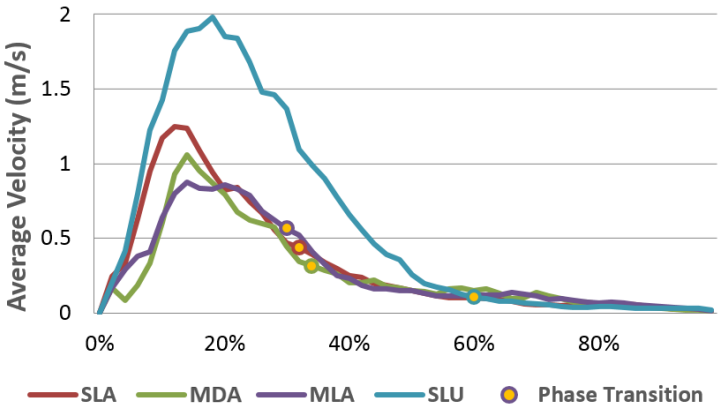


Figure 5.4: Normalized velocity profiles grouped by variations (circles depict t_{ph}).

and they needed extra care to perform the task precisely. It is interesting to notice that, on the contrary, group MLA (6.1) expressed a level of confidence similar to SLU (6.0) and higher that SLA (5.1), while their real averaged accuracy level was similar to MDA and SLA (around 6.0cm of precision).

To support these findings we performed a between subjects One-Way analysis of variance (ANOVA) to compare the effect of stereo vision, avatar and display size on the participant's performances (expressed by the following dependent factors: t_e, t_{ph}, t_{phr}, d_e, d_{ph}, v_{avg}, a_{avg} during phases M_{ap} and M_{ad}) in SLA, SLU, MLA, and MDA conditions. The test for normality, examining standardized skewness and the Shapiro-Wilks test, indicated the data to be statistically normal. The Homogeneity test (Levene's test) also reported non–significant variance between the groups. An alpha level of .05 was used for the analysis and the post-hoc analysis was performed using a standard Tukey-HSD test for comparison of means and Bonferroni's correction.

The analysis showed that there was a statistically significant difference between groups for several factors. Considering the averaged time t_e to reach each target ($F(3,36) = 15.909, p < .001$), averaged velocity v_{avg} ($F(3,36) = 22.169, p < .001$), averaged acceleration a_{avg} ($F(3,36) = 11.428, p < .001$), phase transition time t_{ph} ($F(3,36) = 6.992, p < .001$) and relative t_{phr} ($F(3,36) = 12.46, p < .001$), SLU resulted in faster and shorter motions with respect to the other three variations (Means and Standard Deviations are reported in Table5.3). Considering the distance to the target d_e at the end of the averaged trials ($F(3,36) = 15.253, p < .001$) the

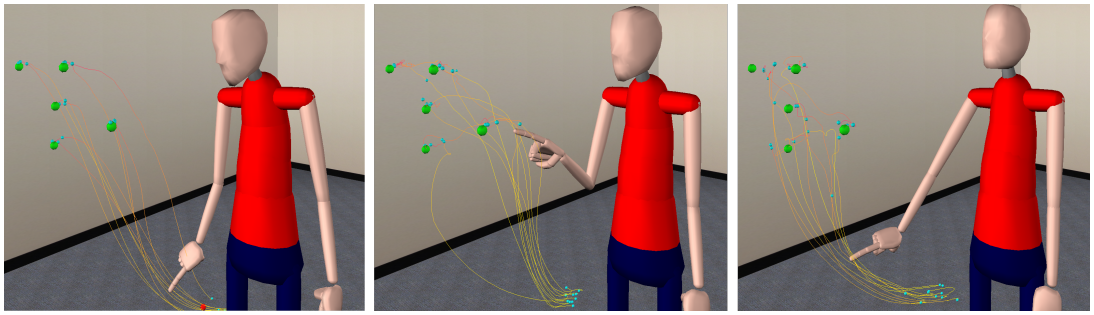


Figure 5.5: Example trajectories collected from one participant in experiment 1. Large (green) spheres represent the targets, and small (blue) spheres represent the start, the end, and point t_{ph} in each trajectory. The show trajectories, in left-right order, were collected in configurations SLU, SLA and MLA.

ANOVA showed that there is a significant separation between the performances of users under stereo vision (SLU: Mean $(M) = 0.005$, Standard Deviation $(SD) = 0.004$; SLA: $M = 0.029$, $SD = 0.04$) with respect to mono vision (MLA: $M = 0.100$, $SD = 0.04$; MDA: $M = 0.080$, $SD = 0.31$). Similarly, the same subdivision could be found already during the transition phase d_{ph} $(F(3, 36) = 16.974, p < .001)$.

Even though the subdivision between the use of large visualization system versus the desktop setup seemed to be affecting performances, it could not be significantly stated.

5.3.3 Discussion

Based on the results, our observations, and on the comments of the participants we have drawn several inferences from the evaluation of reaching accuracy.

As expected, users were faster and more accurate when in user-perspective vision with direct interaction (SLU). Users were 6 times more accurate and 2 times faster than the second best case scenario (SLA). In addition, participants were 4 times closer to each target already at the end of the approaching phase and consequently they needed less time to reach each target. These numbers indicate that the use of the avatar increased the cognitive load of the participants, since they had to transfer their arm movement attention to avatar space. Considering these findings we can conclude that direct interaction with user-perspective stereo vision is a better choice for precision tasks that depend on environment constraints such as reaching for a target.

We can also observe a significant difference in accuracy among the configurations employing avatars. Users were 3 times more accurate with the use of stereo vision than in mono visualization. In addition, users also approached targets more accurately (20% closer), resulting in adjustment motions more focused around the target center area. The execution times and the overall execution velocities were similar across the three variations using the avatar. User perspective stereo vision seemed to improve the space understanding even when the task is transferred to avatar's space.

With respect to the difference between small and large displays we cannot state any significant conclusion. The data however show a trend towards the conclusion that the groups using large displays do not gain any benefits in terms of accuracy, while their perception of being accurate can be slightly compromised. Participants performing reaching tasks in avatar space with mono vision and small display (MDA) were, on average, 10% slower than users performing the same task using a large display (MLA). Similarly, the trajectories generated were 7% longer during the adjustment phase and participants using the small display showed an increase in precision of about 25% in comparison with the users using the large display. They also reported that the task performed in this condition was uncomfortable and they needed extra care and attention (3.6 out of 7.0). On the contrary, participants using the large display felt overconfident on judging their performances (5.2 out of 7.0). They believed to have precisely reached targets, and spent less time during the adjustment phase, resulting in a less accurate final position.

Besides the discomfort (small display users) and misestimation (large display users) acknowledged in the post experiment questionnaire, several other factors, due to the physical setup and hardware choices, may have influenced the participants. For example, even though the resolution between the large and small display is comparable and the distance to the screen has been chosen to provide a similar interaction setup this choice induces a different field of view as well as pixel perception between the two media (pixel size and distance to the screen). In addition, the projection system used for the experiment has a slightly worst performance in terms of pixel sharpness, overall contrast and luminosity in comparison to a high profile desktop LCD monitor (that becomes even worst when applied to back-projected surfaces).

Although these details may have slightly induced participants using the large display to inaccurately perceive the target, our findings are in accordance from results presented in previous works proposed by [64] and [294] where large screens did not improve task performance of users analyzing Three-Dimensional DRMI or navigation of geographical maps. Conversely, reaching interactions, as well as image analysis

and interaction, are affected by display size differently than other types of interactions. For example, Ni et al. [244] have reported that large displays improve user performances during navigation and element search in virtual environments. In fact, the main difference between reaching interactions, image analysis and map navigation is the lack of situation awareness. A deeper evaluation should be performed in order to isolate and possibly quantify how different types of interactions are affected by different screen sizes.

We have also investigated the visual aspects of the generated trajectories and body motions. Trajectories generated using the virtual pointer were smooth with a typical "S-shape" defined by the user raising and approaching the target from a frontal point (Figure 5.5-left). In most cases the user did not pay attention to the environment and intersections with the supporting virtual blue plane would often occur during the approach phase.

Considering the trajectories generated from the avatar motions, we noticed that SLA resulted in more natural (human-like) motions. Participants paid more attention to driving the fingertip of the character towards the target from a frontal position and carefully avoided the supporting surface (Figure 5.5-center). The pattern observed was: users first raised their hands with an elbow flexion to naturally avoid the virtual plane, and then approached the target from the front. In the variations adopting mono vision (Figure 5.5-right), on average, participants did not consider the virtual space occupied by the avatar. The observed pattern was: the avatars arm was first raised to the target's height, without bending the elbow, the avatars hand was then retracted until the fingertip was in front of the target, and then adjusted towards the target position. A mid-term result can be observed in configuration SLU (Figure 5.5-left).

These observations show that the coupling of avatar and stereo vision was optimal in having users pay attention to the upper-body motion displayed by the avatar. Users made the effort to produce a realistic motion instead of simply focusing on maneuvering a pointer to reach targets.

5.4 Experiment 2: Motion Reproduction

The second experiment investigated if and how avatars and user-perspective stereo vision affected the spatial understanding of motions to be reproduced. The experiment had two phases: the demonstration phase and the reproduction phase. In the demonstration phase a blue virtual character (the "tutor") appeared in front of the user and demonstrated a pre-defined upper-body motion. Later in the reproduction

phase the user was then asked to reproduce the observed motion (see Figure 5.6).

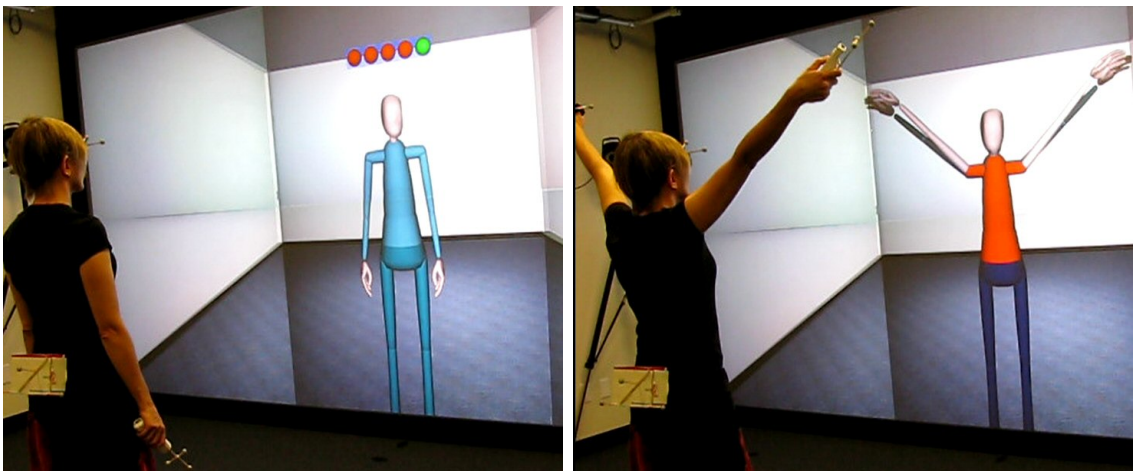


Figure 5.6: Experiment 2 investigated the influence of avatars and stereo vision during motion observation (left image) and reproduction (right image).

Before each motion demonstration participants were instructed to memorize the motion they were going to observe, and to pay attention to details like: motion speed, arm key poses, final height of the hands, and torso orientation. The participants were allowed to watch the demonstrated motion up to three times but they were not allowed to move and simulate the task with their bodies.

The demonstrated motion, or reference motion (M_r), was designed to be simple and not ambiguous. It consisted of three arm raises. Each raise started by raising both arms simultaneously from the rest posture until the hands surpassed the head, with the elbows straight, and then the arms would return to the rest posture. First a lateral raise with arms parallel to the coronal plane was performed, then followed a frontal raise parallel to the tutor's sagittal plane, and then followed a raise exactly in-between the lateral and frontal raises.

During the reproduction phase the participants then reproduced the motions together with the virtual tutor. When the user's avatar was employed, a red avatar was displayed mimicking the user's motions in real-time as in a virtual mirror. In this case the tutor avatar was rendered transparently, overlapping (in fact slightly behind) the user's avatar. The tutor and avatar motions were still clearly distinguishable. Figure 5.6-right shows both avatars being displayed, with the tutor's arms visible slightly below the arms of the user's red avatar.

In order to prepare the participants to promptly start the reproduction fairly

Table 5.4: Configurations of experiment 2 (G = Group, M = Male, F = Female).

Label	G	M	F	Stereo	Avatars
SLA	**4**	5	5	*yes*	*user's avatar and virtual tutor employed*
SLT	**1**	3	7	*yes*	*only tutor employed, no avatar*
MLA	**2**	4	6	*no*	*user's avatar and virtual tutor employed*
MLT	**5**	3	7	*no*	*only tutor employed, no avatar*

in sync with the tutor, a five seconds traffic light was displayed. The participants were informed that the virtual tutor would start to move immediately after the green light.

Similarly to the previous experiment, forty participants took part on the experiment and it was performed in four variations (10 participant per variation), as described in Table 5.4. The variations covered the joint combination of employing or not the avatar and the user-perspective stereo vision.

5.4.1 Data Collected

Similarly to experiment 1, the full motions performed by the participants were recorded for each variation. For each motion, we extracted the trajectories generated by the wrist joints in global coordinates. We denote t_e as the duration of a trajectory (motion) in time, and t_{p1} and t_{p2} as the time values that divide the trajectories in the three distinctive motion phases: lateral raise M_l, frontal raise M_f and intermediate raise M_i. Values t_{p1} and t_{p2} were manually annotated for each motion and in the reference motion.

The performances were analyzed in terms of time, distance and velocity differences when comparing each recorded trajectory against M_r, and after aligning the trajectories according to each cyclic phase. The time difference per phase, here denoted as phase synchronization time, was calculated by subtracting t_{p1}, t_{p2} and t_e from their M_r counterparts. The obtained differences are denoted as t_{d1}, t_{d2} and t_{de}.

Distance and velocity errors were calculated using the following procedure: each performed trajectory was subdivided and time-aligned (by uniform time warping) with the corresponding segments M_l, M_f and M_i of the reference motion. Each trajectory segment was also uniformly re-sampled. We denote the re-sampled trajectories as S_l, S_f and S_i. Each trajectory was compared with the corresponding samples in M_r in order to achieve the error values. Distance errors between corresponding samples are denoted as S_{dl}, S_{df} and S_{di}; and velocity errors are denoted by S_{vl}, S_{vf} and S_{vi}.

Table 5.5: Experiment 2 descriptive statistics summary. Averaged groups performances: t_{de}, t_{d1} and t_{d2} denote the end of phase difference in time between the user's motion phase and the reference motion phase M_r; S_{vl}, S_{vf} and S_{vi} denote the average velocity and S_{dl}, S_{df} and S_{di} denote the distance error per phase from the reference motion M_r. The shown values are mean values with the standard deviation in parenthesis.

	SLA	**SLT**	**MLA**	**MLT**
phase sync. (s)				
t_{d1}	0.384 (0.297)	0.702 (0.486)	0.568 (0.490)	1.591 (0.963)
t_{d2}	0.533 (0.191)	0.836 (0.689)	0.908 (0.441)	2.320 (1.232)
t_{de}	0.396 (0.280)	1.243 (0.973)	1.167 (1.037)	2.923 (2.076)
average	0.437 (0.114)	0.927 (0.548)	0.881 (0.531)	2.278 (1.199)
vel. error (m/s)				
S_{vl}	0.118 (0.069)	0.246 (0.068)	0.210 (0.128)	0.210 (0.162)
S_{vf}	0.080 (0.056)	0.089 (0.061)	0.163 (0.126)	0.149 (0.122)
S_{vi}	0.272 (0.078)	0.308 (0.143)	0.247 (0.114)	0.347 (0.166)
average	0.156 (0.039)	0.214 (0.073)	0.206 (0.095)	0.235 (0.135)
dist. error (m)				
S_{dl}	0.174 (0.055)	0.173 (0.069)	0.243 (0.174)	0.283 (0.213)
S_{df}	0.207 (0.079)	0.208 (0.047)	0.296 (0.124)	0.301 (0.139)
S_{di}	0.216 (0.065)	0.228 (0.083)	0.371 (0.205)	0.294 (0.163)
average	0.197 (0.037)	0.203 (0.034)	0.305 (0.154)	0.292 (0.146)

Table 5.6: Experiment 2 descriptive statistics summary. Factor combinations: t_{de}, t_{d1} and t_{d2} denote the end of phase difference in time between the user's motion phase and the reference motion phase M_r; S_{vl}, S_{vf} and S_{vi} denote the average velocity and S_{dl}, S_{df} and S_{di} denote the distance error per phase from the reference motion M_r. The shown values are mean values with the standard deviation in parenthesis.

	stereo	**mono**	**avatar**	**tutor**
phase sync. (s)				
t_{d1}	0.543 (0.425)	1.079 (0.910)	0.476 (0.406)	1.146 (0.871)
t_{d2}	0.684 (0.516)	1.614 (1.156)	0.720 (0.383)	1.578 (1.234)
t_{de}	0.819 (0.821)	2.045 (1.892)	0.781 (0.958)	2.083 (1.798)
average	0.682 (0.460)	1.579 (1.159)	0.659 (0.455)	1.602 (1.142)
vel. error (m/s)				
S_{vl}	0.182 (0.093)	0.210 (0.142)	0.164 (0.111)	0.228 (0.122)
S_{vf}	0.084 (0.057)	0.156 (0.121)	0.121 (0.104)	0.119 (0.099)
S_{vi}	0.290 (0.114)	0.297 (0.148)	0.259 (0.096)	0.327 (0.152)
average	0.185 (0.064)	0.221 (0.114)	0.181 (0.075)	0.224 (0.106)
dist. error (m)				
S_{dl}	0.173 (0.061)	0.263 (0.190)	0.208 (0.130)	0.228 (0.164)
S_{df}	0.207 (0.063)	0.298 (0.128)	0.254 (0.112)	0.252 (0.111)
S_{di}	0.222 (0.073)	0.332 (0.185)	0.293 (0.168)	0.261 (0.131)
average	0.200 (0.035)	0.298 (0.146)	0.251 (0.122)	0.247 (0.113)

Since the reference motion was designed to be simple and symmetric, as expected, the difference between the left and the right samples was not significant. For this reason we omit the disjoint analysis between the left and right samples and report only the averaged left-right investigation.

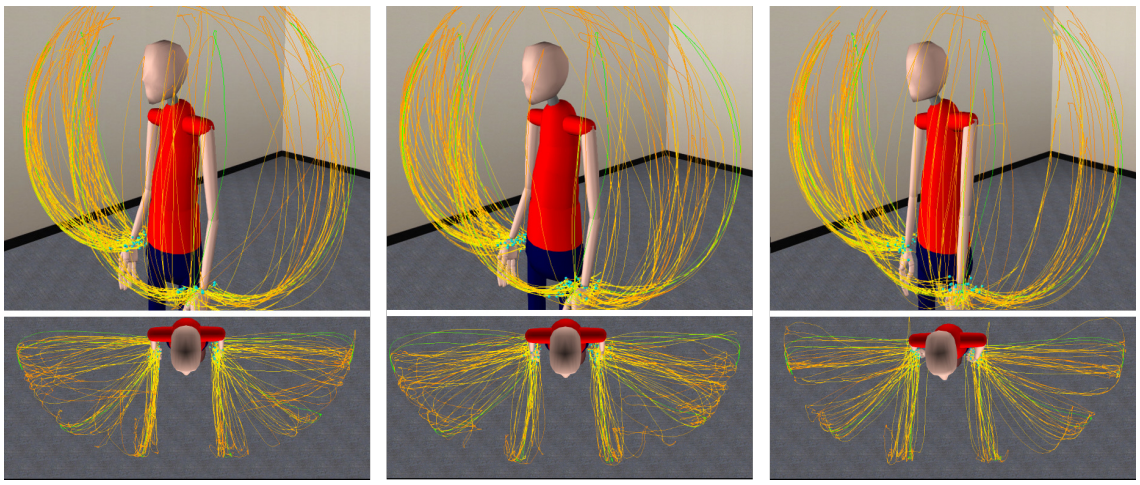


Figure 5.7: Trajectories collected from the character's motions during experiment 2. The green trajectory shows the reference motion M_r. The small blue spheres represent points t_{p1} and t_{p2}. Left images: SLT shows fairly separated phases but the heights of the trajectories did not well correspond to the heights in M_r. Center images: MLA shows merged trajectories between phases M_i and M_l. Right images: SLA shows the best results, with the separation space and height traversed by each phase being closest to M_r.

5.4.2 Results

In our initial descriptive statistics we averaged the participant's readings by category (Table 5.5 - Variations) and performed a joint analysis between the independent categories (Table 5.6 - Factors).

From the grouped variations table (Table 5.5) we can infer that participants in the variations with the avatar (MLA and SLA) showed a better phase synchronization than users in the other variations. This effect is also reflected by the velocity error and partially by the distance error. In this last case, SLA still shows the best result while MLA users were less accurate. Similarly we can infer the same pattern from the joint factors sub-table (Table 5.6) where, except for the distance error, during variations

using avatars and variations using stereo vision, users had the best performances (time phases synchronization and less velocity profile error).

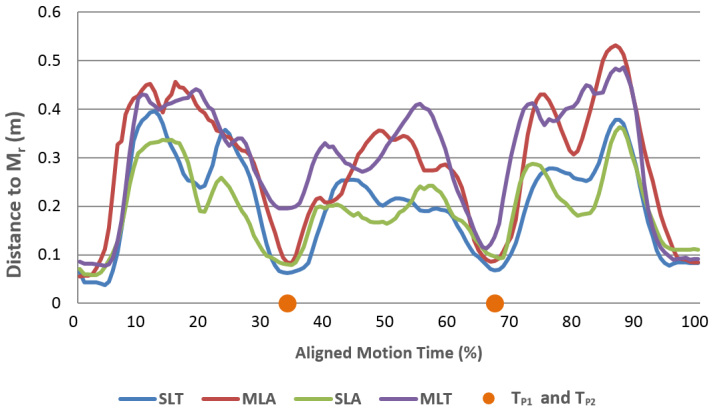


Figure 5.8: Aligned distance profiles of replicated motions in each variation (orange dots depict t_{d1} and t_{d2}).

This effect can also be observed from generated trajectory depicted in Figure 5.7. Looking at the trajectories generated in variation SLA, the compliance to the reference motion is maintained similarly to the case of using stereo but without the avatar (SLT), however the height and key poses were better respected in SLA.

Analyzing the average distance compliance with M_r we can infer that participants using the system in mono vision performed in the least accurate fashion, $10cm$ difference on average per section. Figure 5.8 shows in detail the averaged distance profile of each variation highlighting each phase. When stereo vision was used the motions were closer to the reference motion, in particular, phase M_i (arms raising diagonally) was better perceived by the participants. The error from participants in MLA and MLT is clearly higher that the error produced by participants performing in SLT and SLA during M_i. On the contrary, if we consider the factors combination sub-table this subdivision disappears. An explanation for this behavior can be given considering the joint performance of mono vision and without the avatar, which represents a non-optimal configuration.

Considering the velocity profiles (Figure 5.9) we notice that both variations showing the avatar were overall closer to the reference motion profile than the other variations. It is also interesting to notice that in all variations participants anticipated the start of the first phase. This fact might have been driven by the implementation choice of using the virtual traffic light, as they felt driven to start right after the

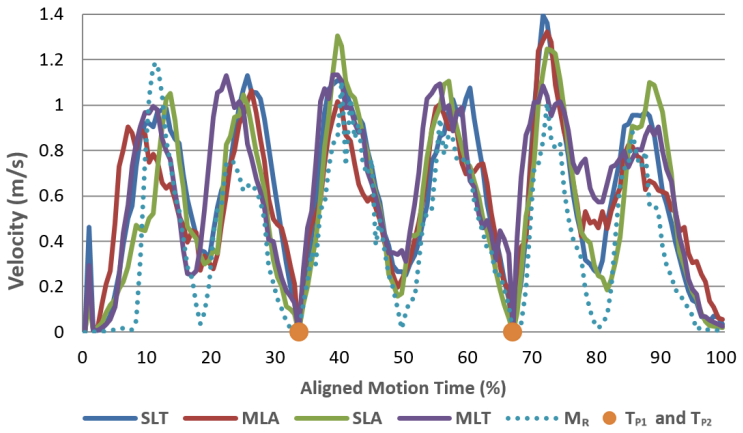


Figure 5.9: Aligned velocity profiles of reference and replicated motions in each variation (orange dots depict t_{d1} and t_{d2}).

green light appeared instead of waiting for the virtual tutor to start the motion.

From the data collected in the post-experiment questionnaire participants equally rated, in all the variations, their confidence of having perceived the motion correctly. In terms of reproduction, participants under the conditions without the avatar felt to be slightly less accurate in terms of position and speed (on average 15% less). Conversely, the presence or absence of stereo vision did not affect the users' level of confidence.

After this preliminary study a two-factor (2x2) ANOVA (SS Type III) was performed to evaluate the effect of the visualization type and the presence of the avatar on the user performances represented by each accuracy measures: t_{d1}, t_{d2}, t_{de}, S_{dl}, S_{df}, S_{di}, S_d (average of the previous three values), S_{vl}, S_{vf}, S_{vi}, and S_v (average of the previous three values). The test for normality, examining standardized skewness and the Shapiro-Wilks test indicated the data to be statistically normal. In order to meet the Homogeneity assumptions for performing a two-way ANOVA the data were transformed using a standard natural power transformation (p-values reported below resulted from not significant tests for homogeneity of variance). The reported estimated means have been back-transformed to reflect the original data and the standard deviation has been reported as interval (SDI) due to the non-linear back transformation. An alpha level of .05 was used for the initial analysis.

Considering the distance compliance with the reference motion M_r, the results for the two-way ANOVA indicated a significant main effect for the visualization type, per phase (S_{dl}: $F(1,36) = 5.755, p = .022$; S_{di}: $F(1,36) = 7.360, p = .009$) and overall

(S_d: $F(1,36) = 10.588, p = .002$). A review of the group means for the averaged distance factor (S_d) indicated that the error of the group using user-perspective stereo vision ($M = 0.197$ $SDI = [0.175, 0.221]$) had a significantly lower level of error than the group interacting without stereo ($M = 0.258$ $SDI = [0.229, 0.291]$). The analysis confirmed our initial conclusions and we can state that user-perspective stereo vision resulted in motions with significantly higher compliance to the reference motion.

In addition, we examined the participants capability of respecting key poses in time with the reference motion M_r. Both visualization type and the presence of the avatar (disjointly) showed main effects. The visualization type produced a main effect per phase (t_{d1}: $F(1,36) = 5.755, p = .022$; t_{de}: $F(1,36) = 9.280, p = .004$) and overall ($t_d$: $F(1,36) = 21.016, p < .001$). Considering the averaged estimated means we can infer that participants in user-perspective condition better respected key times ($M = .654$ $SDI = [.345, .936]$) with respect to the other groups without ($M = 1.475$ $SDI = [1.184, 1.767]$). Similarly, the presence of avatar, also produced main effects per phases (t_{d1}: $F(1,36) = 6.870, p < .013$), t_{de}: $F(1,36) = 16.416, p < .001$) and overall ($t_d$: $F(1,36) = 24.942, p < .001$)). The presence of avatar ($M = .555$ $SDI = [.264, .847]$) helped the user to better respect the original motion key poses with respect to time in respect to the groups without the avatar ($M = 1.565$ $SDI = [1.274, 1.856]$).

Considering the joint factor analysis, in phase t_{d1} and t_{de}, the effect trended toward significance (t_{d1}: $F(1,36) = 3.225, p < .080$ and t_{df}: $F(1,36) = 3.710, p < .062$). Similarly S_{dl} and S_{df} also trended toward significance (S_{dl}: $F(1,36) = 3.449, p < .071$ and S_{df}: $F(1,36) = 3.527, p < .068$). Looking at the estimated mean comparisons it seems that avatar use (both in time and distance) improved the reproduction of motions when only mono vision was used. However, since there is no statistical significance and the interaction appeared only on subsets of the data, the main estimates have not been dropped.

Finally, considering the velocity profile maintenance ANOVA reported only a trend on the presence of the avatar main effect on the phase S_{vl} ($F(1,36) = 3.449, p < .071$). Even though this conclusion could not be supported statistically, by analyzing the motions visually it is possible to notice that in the presence of the avatar participants reached more accurately (in terms of distance and timing) peaks and key points, resulting in small partial velocity errors. Changes in acceleration could be however noticed in a few trials, due to users assessing the virtual character's motion and trying to catch up with the tutor. The motions appeared less fluid in these cases.

5.4.3 Discussion

Our study provides evidence that both the use of avatars and stereo vision positively affect motion reproduction tasks.

The presence of avatar and stereo vision both improve the ability to preserve spatial alignments when reproducing motions. Training applications represent a typical scenario where both these conditions would be useful. Accurate motion perception is particularly important when demonstrated motions have relationships with the environment or objects, in which cases key poses will have to be well perceived in order to achieve effective training.

We can also observe that the avatar helps driving the improvement of the motion reproduction when stereo vision is not employed. Considering a scenario where motion reproduction is performed without stereo vision, displaying an avatar is a key recommendation. This is for instance the case of several applications related to delivery of physical exercises at home, where it is difficult to implement stereo vision.

5.5 Experiment 3: Motion Modeling

The third experiment investigated the usability of a complete interface for immersive motion modeling by direct demonstration (see Figure 5.10). In this experiment participants were asked to perform motions to be performed by a virtual character.

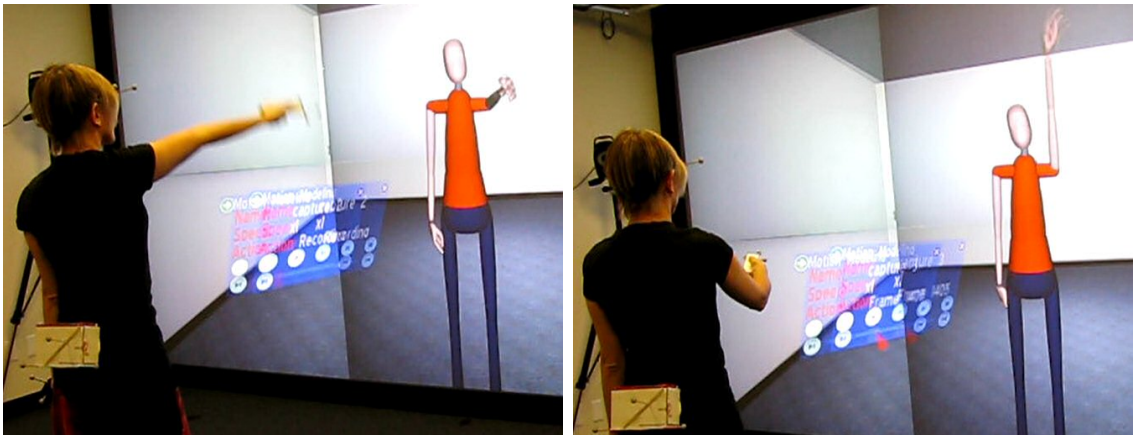


Figure 5.10: Experiment 3 evaluated an immersive motion modeling by demonstration interface. Motion recording with or without avatar (left image) and playback (right image) were evaluated.

Two variations of the system were considered by varying the use of the avatar, as summarized in Table 5.7. Given the focus on investigating the usability of the system, and that detailed trajectory analysis was already explored in the previous experiments, only questionnaires were used for evaluation.

Table 5.7: Configurations of experiment 3 (G = Group, M = Male, F = Female).

Label	G	M	F	Description
SLN	**5**	3	7	*Recording phase without the avatar*
SLA	**4**	5	5	*Recording phase with the avatar*

Similarly to experiment 2 the scenario included a blue virtual tutor character placed in front of the user. The tutor was used only at the beginning of the experiment to demonstrate to the user an upper-body motion composed by a sequence of simple gestures (arm raises, elbow bends, arm pointings, etc.). At this stage the participant was required to memorize the overall motion until satisfaction, being allowed to move and replicate the motion physically to help with the memorization. The sequence of motions included both the left and the right arm.

After this preliminary step the user was asked to use the virtual user interface to model the motion previously observed. The motion was modeled by direct recording via the motion capture markers. The tutor character was hidden in all variations. In variation SLA, the red user's avatar was displayed during recording. In variation SLN the avatar was hidden during motion recording, and the participant was required to record motions without any visual feedback.

The interaction with the graphical interface consisted of a virtual pointer floating in front of the user's hand (in front of the Wii-mote controller). Two buttons from the controller were used to select actions from virtual panels and to show/hide the interface at will. Since the implementation used user-perspective stereo vision the interface was perceived as floating in front of the user and it could be repositioned in space at any time to avoid occlusions with the scene or to be better approachable. The interface provided tools for recording, playing, trimming, and inspecting specific portions of the recorded motions. The user could also model and store more than one motion until satisfaction, with the possibility to re-select or discard motions.

5.5.1 Data Collected

At the end of each trial, a questionnaire was administered about the usability, preferences, user experience, and also asking for suggestions for improving the interface

and the overall approach. Except for a few open questions requesting feedback, the questionnaire consisted of seven-point Likert-scale items. See Table 5.8 for an excerpt of the questionnaire. The full motions saved by the participants were also stored but they were only used to validate if the users performed all the motions required to be modeled.

5.5.2 Results and Discussion

Since the task of modeling motions by demonstration required to handle a more complex interface and implied more steps and familiarization with the system, as expected, the task was rated by the participants to be more difficult than the tasks in the previous experiments. Looking at the control questions, when asked about confidence on completing the task correctly and confidence on being able to coach the task to someone else, we observed a 15% decrease of confidence with respect to experiments 1 and 2. In experiments 1 and 2 the average level of confidence was 6.4 out of 7, while in Experiment 3 it was 5.7 out of 7.

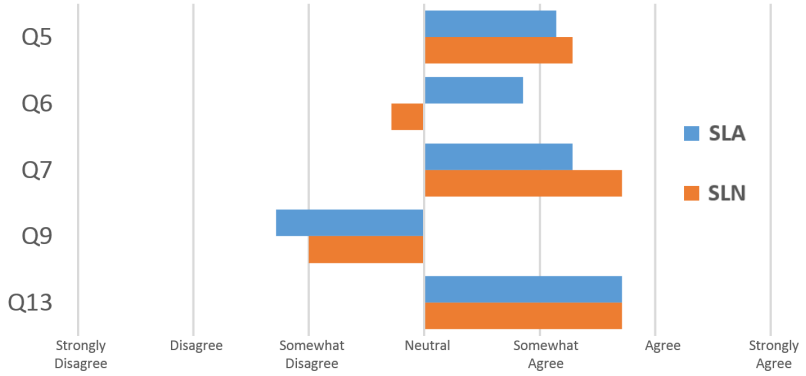


Figure 5.11: Results from selected usability questions for Experiment 3. The corresponding questions are available in Table 5.8.

Considering the targeted questions and the open comments the users positively rated the overall experience. The average rating for the question "Rate how comfortable you felt during the performance in all the aspects of the activity (1=extremely uncomfortable, 7=extremely comfortable)" was 5.5.

Figure 5.11 summarizes results of selected questions from the post-activity questionnaire. Q6 highlights the fact that users in SLN wanted additional training before engaging in the motion modeling activity. Comments and suggestions also noted that they did not know how to behave when the recording phase started. Before the be-

ginning of each recording phase the system warned the user with a timer countdown with textual messages, but this was not sufficient for them to grasp the activity in their first attempt. After a few trials they were then able to record motions correctly. This fact was also reflected by question Q14 of the SLN questionnaire, where users expressed the hypothesis that having an avatar during the recording session would have improved the usability of the system (6.29 out of 7). Users performing in SLA felt that the avatar helped them to better understand their motions (Q15-SLA 6.57 out of 7) and they did not feel distracted by it (Q14-SLA 6.57 out of 7). These results are consistent with experiments 1 and 2.

Table 5.8: Extract of the questionnaire administered in Experiment 3 (likert-scale rated between 1=strongly disagree, 7=strongly agree).

Q5	The interface was simple and easy to understand.
Q6	The interface could be used without training.
Q7	3D vision is important to model motions.
Q9	I would have preferred to use a standard computer.
Q13	The approach to model motions was effective.
Q14*	The avatar was distracting while recording.
Q14**	Seeing my motions while recording would have helped me.
Q15*	The avatar helped me to be more precise.
	(*SLA only, **SLN only.)

5.6 Conclusions

The presented experiments have uniquely studied the effectiveness of avatars and user-perspective stereo vision during task performance. Our results have shown the viability of the approach of direct motion demonstration for modeling and reproducing motions. The correct use of avatars was shown to have great potential to improve performances in a number of situations; however, we have also observed that there are critical design choices that highly influence the suitability of the configurations to different types of interaction needs.

Our experiments confirm that the use of user-perspective stereo vision with direct interaction is the optimal choice in terms of task accuracy and completion time, when precision tasks are involved. Direct interaction made users 6 times more accurate and 2 times faster than in other conditions. For example, in the widely adopted

scenario of stroke rehabilitation, tasks often involve repeatedly reaching regions in space. In such cases user-perspective stereo vision will lead to better accuracy when measuring the rehabilitation progress.

User-perspective stereo vision also improved replication of spatial relationships even when the task was transferred to the avatar's space (by a factor of 3). When the task involved motion reproduction, stereo vision showed improvements both in terms of synchronization and compliance with the reference motion. The use of avatars produced increased attention to the avatar space, allowing users to better observe and address motion constraints and qualities. Coupling avatar use with stereo vision resulted in users paying more attention to the motions within the virtual environment, improving the realism and correctness of the motions. These findings represent key factors to consider when designing applications for distant training in collaborative spaces where the kinesthetic component is fundamental with respect to the environment.

In addition, avatar use was shown to improve motion reproduction in the cases where stereo vision was not present. This factor suggests that in application setups where stereo vision is not practical to be used (such as in homes or clinics), the use of avatars will definitively improve the user understanding of motions displayed by virtual tutors.

In summary, if the involved tasks require generation or reproduction of motions with desired qualities, such as in training applications where the environment or objects and tools are key factors, the use of avatars and stereo vision will improve that ability. However, if the goal is to accomplish tasks no matter the type of motions used, direct interaction in user-perspective will be more efficient. Besides the additional instrumentation (stereo glasses), stereo vision was shown to be always beneficial.

The above conclusions were found to be statistically significant and additional important trends and observations were also made. We have noticed that the small display induced users to not overestimate their capabilities during precision tasks. However, the small display increased their execution time and frustration. Although further investigation should be performed to fully support this theory, a new guideline on designing gesture-based game controllers can be drawn. Given that the dimensions of the target visualization system was observed to affect user expectations, game controller precision and other difficulty settings should be dynamically adjusted with respect to the display size in order to manage user frustration.

Overall, the presented results provide important new quantification and observations in each of the performed experiments, leading to a new understanding of the

trade-offs involved when designing avatar-based training systems.

CHAPTER 6

Application to Physical Therapy

Physical therapy is a broad field that addresses the recovery and treatment of injuries, physical impairments, disabilities, diseases and disorders related to motor and balance dysfunctions affecting many daily life activities. A rehabilitation process is usually necessary for patients after a specific type of injury involving physical (impingement, surgery, arthritis, etc) or neurological (strokes, neuropathies, etc) impairments.

Rehabilitation and physical therapy are optimal when assessment, monitoring, patient engagement, and adherence to the therapy program can be achieved. In traditional physical therapy different processes are usually involved: physical examination, evaluation, assessment, therapy intervention, monitoring, and modification of the therapy program according to patient recovery [44].

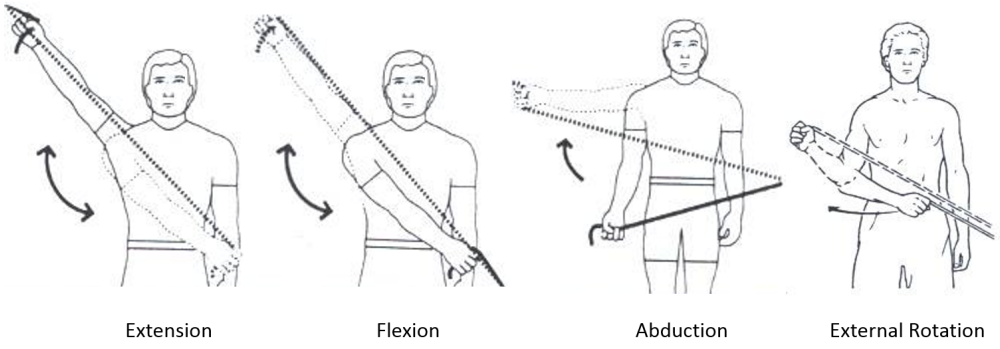


Figure 6.1: Example of a typical paper description of an exercise given to patients.

While it is clear that VR-based computer systems for therapy delivery have great potential to well address most of the limitations of traditional physical therapy (e.g. lack of customization, engagement, supervision, etc.), implementing effective solutions involves multiple challenges. In any case, current practices can be certainly

improved. For example, Figure 6.1 illustrates how exercises are typically described in paper to patients when they are given a set of exercises to be executed at home. Paper descriptions suffer from perceptual limitations and lack of interactivity, and mostly important they do not provide monitoring and logging capabilities that are crucial for determining patient adherence to the program and the effectiveness of the exercises.

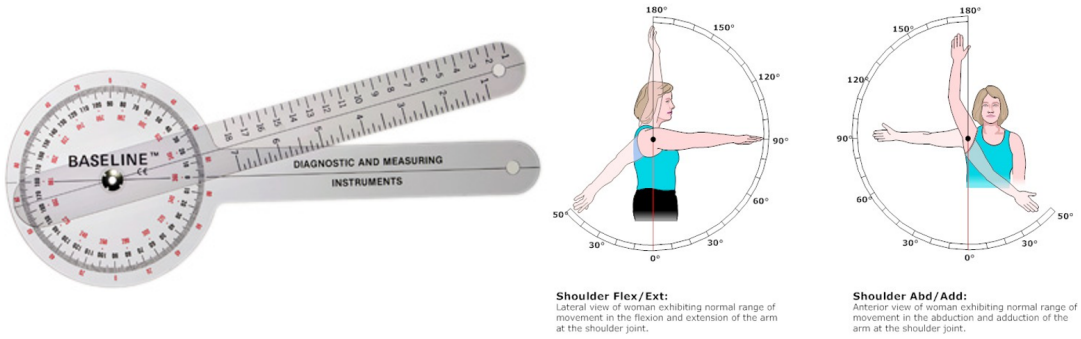


Figure 6.2: Example of a typical tool used to measure improvements and paper representation of the measurement.

In traditional physical therapy even the measurements of patient improvements and monitor is done using mechanics tools (Figure 6.2)

Our approach addresses these challenges in a unified way. We first design motion demonstration methodologies that allow therapists to intuitively create, edit and re-use customized exercises that are responsive to the needs of their patients. In this way we integrate in our systems the ability to configure exercises to particular patients, both in terms of creating new exercises as needed and in terms of designing how exercises should adapt to patient preferences, physical limitations, and recovery rates. In addition, several factors can be considered to adjust a system to the user's preferences: from the language and pace to display messages and instructions, to the appearance of the virtual character demonstrating the exercises, etc. Multiple interaction channels can be customized in order to approach a similar set of communication channels that the patient is used to experiencing during his or her daily human-human interactions. Cultural background, social and age groups all play important roles in the wide variation of preferences that can be identified and modeled in VR systems.

We describe in this chapter our first steps towards such an adaptive and responsive interactive therapy system, discussing adaptive exercise models, motion processing algorithms, and exercise delivery and monitoring techniques that are able to effectively respond to physical limitations and recovery rates of individual patients. The presented solutions provide a basic framework to experiment and address a first set of adaptation and customization features, and we focus on adaptation of exercises for the shoulder complex. We also present remote networked solutions for allowing therapists and patients to share motion performances in real-time. The transmitted data is lightweight and remote collaboration can well scale to several patients at the same time. The capability of remote sessions is important in order to keep patients motivated and engaged in the therapy when they are supposed to work on their therapy programs at home. Remote sessions also have great potential to reduce costs and to widen health care delivery.

A number of additional features are also presented for achieving a complete framework for therapy modeling, delivery and analysis. Our system provides 3D assessment tools for monitoring a range of motion, and for allowing the visualization of a number of therapy parameters during or after execution of exercises. We have implemented our system in two configurations: a low-cost version based on a Kinect sensor and a high-end version based on the full-scale immersive Powerwall (see Figure 6.3).

We have collected informal feedback from therapists demonstrating that adaptive and responsive exercise delivery improves their willingness to adopt the proposed solutions in their practice.

6.1 Related Work

Over the last decade serious games for rehabilitation have become an important research focus with relevant evidence of benefits [104, 181]. Different types of applications have been developed targeting both specific and broad types of applications [97, 261, 165]. Virtual reality has been successfully applied for rehabilitation of stroke patients [53, 49], and with a different purpose, fitness applications have also emerged from videogame interfaces [187] and other custom-made light devices [61].

Perry et al. [266] described the typical workflow of applications with respect to neuro-rehabilitation. The workflow in clinics follows a cyclic process of treatment planning (generation), execution (delivery) and performance assessment. The traditional physical therapy protocol follows a similar pattern and the same concept can be extended to develop applications for physical therapy.

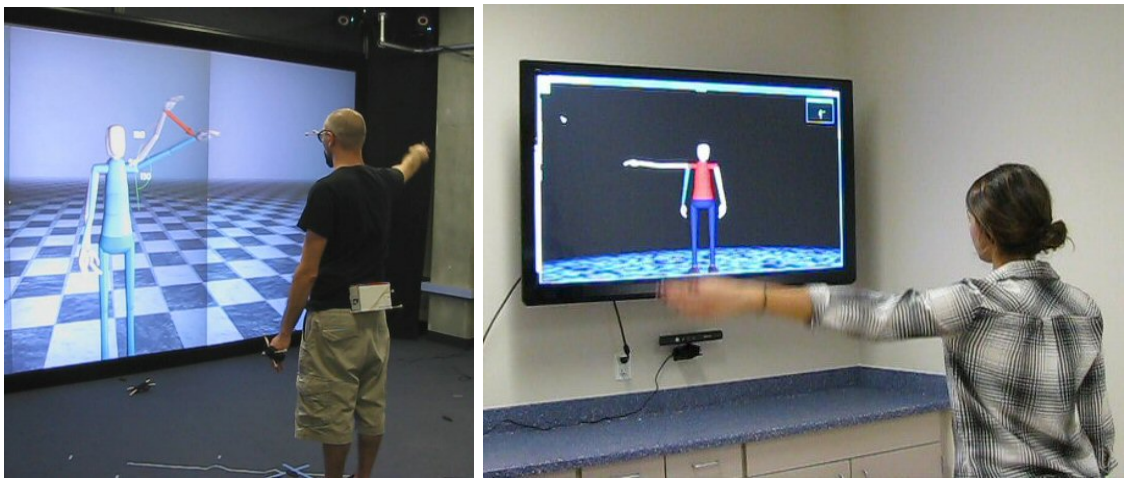


Figure 6.3: Our VR-based collaborative system can run in two configurations: a high-end immersive setup provides improved motion capture and visualization results (left), while the Kinect-based setup provides a low-cost solution suitable for patients and therapists using traditional desktop computers (right). The overlapped virtual characters represent the user's avatar and the autonomous character demonstrating exercises, or the user's avatar and the avatar of the remote participant.

Standard commercial physical therapy packages adopted by clinicians rely on regular media to deliver exercises. The information is usually conveyed through simple text information, sequence of images, and/or video recordings. Users are only controlled and assessed while they interact directly with physicians during in-clinic follow-ups. Patients are therefore many times left unsupervised. The use of new technologies to overcome the limitations of standard approaches to physiotherapy is becoming increasingly popular. For example, the prototype product Reflexion Vera [116] tracks and monitors users through a lightweight sensor (Microsoft Kinect or similar) reporting to the therapist each performance.

Due to the high potential, research relying on the Kinect sensor is being performed to estimate the precision and validity of the device for posture assessment [69] or for motion analysis [92, 40]. Based on these studies, Kinect can be used to reliably track some types of motions, in particular upper-body exercises [206]. Exoskeletons, robotic arms with force feedback and more precise, marker based, tracking systems have also been employed for assisting and monitoring impaired patients; however, involving cumbersome and costly devices is not suitable for widespread use [271, 112, 306].

In our approach the creation and delivery of a physical therapy program follows a programming by direct demonstration strategy. The key benefit is to allow users to intuitively define new exercises as needed. The overall approach has been adopted in many areas [54, 320, 207], and it involves the need to automatically process captured motions according to the goals of the system.

Velloso et al. [360] propose a system that extracts a movement model from a demonstrated motion to then provide high-level feedback during delivery, but without motion adaptation to user performances. The YouMove system [11] trains the user through a series of stages while providing guidance and feedback; however, also without incorporating motion adaptation to user performances. Our approach incorporates motion adaptation in several ways, allowing greater flexibility to achieve effective exercises to patients of different learning abilities, impairments, and recovery rates.

A typical approach for delivering physical therapy exercises is to track user movements while a virtual character displays the exercises to be executed. The representations of both the user and the virtual trainer are usually displayed side by side or superimposed to display motion differences, improving the learning process and the understanding of the movements [126, 377].

Automated systems often allow parameterization capabilities. For instance, Lange et al. [182] describe core elements that a VR-based intervention should address, indicating that clinicians and therapists have critical roles to play and VR systems are tools that must reflect their decisions in terms of taking into account a person's ability to interact with a system, types of tasks, rates of progression, etc. [193, 106]. Geurts et al. [97] describe 5 mini-games that can be calibrated and adapted in terms of speed and accuracy. The physical exercises are static and cannot be replaced. In comparison, our approach is much more comprehensive in that it relies on motion capture and on processing entire full-body motions for adaptation. By doing so we propose new motion processing approaches to achieve adaptive motions that are both controllable and realistic.

Significant research on motion capture processing has been performed in the computer animation field. Motion blending techniques with motion capture data [171, 170, 296, 209] are popular and provide powerful interpolation-based approaches for parameterizing motions; however, they require the definition of several motion examples in order to achieve parameterization. In contrast our proposed techniques are simple and are designed to provide parameterization of a given single exercise motion. We rely both on structural knowledge of exercises and on generic constraint detection techniques, such as detection of fixed points [202, 299] and motion process-

ing with Principal Component Analysis (PCA) [100].

Rehabilitation based on tele-consultation between two healthcare services has been studied with different technologies. In physiotherapy, tele-treatment between healthcare and community services using video has been successfully employed in study cases with elderly with stroke [178] and knee pain [378]. Using virtual reality and serious games, Golomb et al. [103] presented a system for remote rehabilitation of hands for in-home use with distributed data sharing. Several studies have also combined live video of the patient integrated with the virtual environment to augment the patient's feeling of presence in the interactive space [49, 164]. In these applications video was used to provide visual feedback. This choice however does not allow direct interaction in a virtual space. Data collection of a patient performance also becomes a difficult task when users are only captured by regular video.

One development using immersive virtual reality and 3D camera imaging reconstruction has been proposed by Kurillo et al. [177]. This hybrid system allows therapists and patients to share and interact in the same virtual space. The approach however focuses on high-quality rendering and is not suitable as a low-bandwidth solution for physical therapy. An improvement of this work [256] allows the system to additionally detect human poses and assist with balance control. Although remote collaboration has been explored in different ways, a suitable overall solution for interactive sessions has not yet been integrated for remote physical therapy sessions.

We present in this chapter our combined approach to achieve exercises that can be modeled by demonstration, that are responsive to the performances of users, and that can be exchanged in real-time in low-bandwidth remote therapy sessions by limiting transmission to joint-angle data.

6.2 Configurations and Features

We describe in this section the main functionality and configurations that we have developed in our system.

Therapists can design exercises and therapy programs, and then use the system to deliver the exercises in different ways. Created exercises and programs can be stored for further reuse and sharing. When virtual characters autonomously deliver exercises, a number of parameters describing adaptation strategies can be customized, and monitoring and logging tools can be enabled as needed. The provided tools improve patient understanding, motivation and compliance, and also provide data gathering.

Two configurations have been developed, and while the user interface is different the functionality remains the same. Both configurations can work offline, where the patient can only interact with an autonomous virtual (animated) therapist, or online, where remote patients and therapists are tracked simultaneously and their avatars are displayed in the same virtual space. In all cases a number of analysis tools for real-time or post-analysis monitoring, feedback and logging are always available.

The software application has been developed based on the Ogre3D graphics rendering engine [345] and the VR Framework explained in detail in Chapter 3. This choice has allowed us to produce and customize a same application across different modalities and platforms. The system can be easily ported to different operating systems or to more complex virtual reality settings like CAVEs.

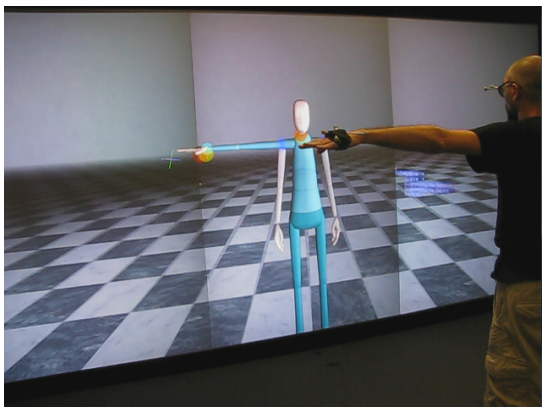
6.2.1 Immersive VR Configuration

Our experimental immersive setup consists of a Powerwall system composed of six rendering computers, a main rendering node and an external computer driving input devices and the motion capture system. The interaction with the application is fully immersive; thanks to virtual pointers and a 3D graphical user interface controlled by a Wiimote. See Figures 6.3-left and 6.4.

The high-end configuration allows therapists to immersively model exercises by demonstration and to experience full-scale visualization of patient performances. The patient's motion can be captured and displayed in real-time or it can be loaded from previously logged sessions. The application provides stereo visualization for enhanced comprehension of the motions and data.

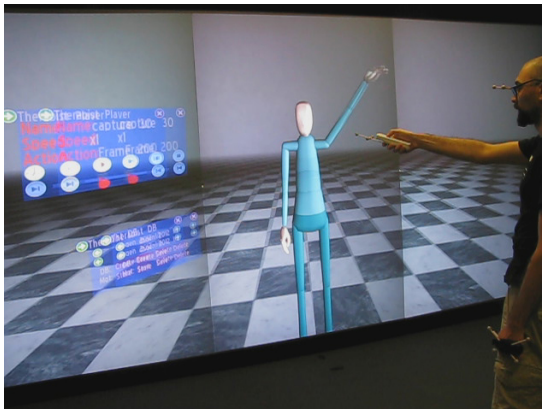
A high-end system configuration also allows the integration of precise tracking capabilities. In our setup the user's upper body motions are tracked using a 10-camera Vicon motion tracking system. For improved usability, the system is configured to only track markers attached to the hands, torso and head. The motion is calibrated and mapped to the avatar following simple scaling procedures as described in the work of Camporesi et al. [54]. This solution has been enough to allow us to experiment with the system; however, since we reconstruct the motion from a reduced marker set not all degrees of freedom of the user's motion can be precisely replicated; in particular, the elbow orbit motion around the shoulder-wrist axis is set to be always in a low-energy position. If precise motion replication is needed, in particular for cases where avoiding compensatory movements is important, additional markers have to be placed on the user.

In remote connection mode the immersive system allows to achieve full-scale

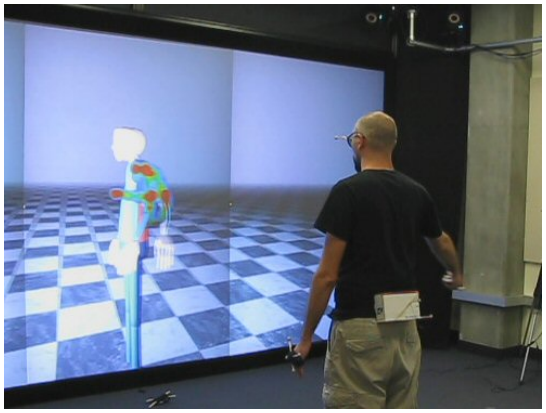


(a) Motion tracking calibration

(b) 3D GUI for recording exercises



(c) Exercise review and parameterization



(d) Exercise analysis

Figure 6.4: Example of using the immersive Virtual Reality configuration.

interactions that are closer to how humans interact to each other. When connected to a remote site, two avatars are displayed for representing the connected patient and therapist. Previously recorded sessions can also be played on any of the avatars. The avatars can be visualized side-by-side or superimposed with transparency.

6.2.2 Low-Cost Configuration

The low-cost configuration is designed to be of simple installation and maintenance at clinics or at home. The patient is tracked through a markerless motion tracking device, in our case using a Microsoft Kinect sensor or similar. Such configuration

is important because it is simple, portable and suitable for any kind of desktop environment. It is also suitable to assist patients in their daily routines in clinical environments. See Figures 6.3-right, 6.8 and 6.16 for examples.

Even though the accuracy of Kinect is limited (the accuracy drops highly when users are not facing the camera or when body occlusion occurs) it still provides a good balance between precision, cost and portability [257, 69, 93, 228, 247].

Kinect-based configurations can also be remotely connected to other instances of the system for collaborative sessions.

6.2.3 Remote Collaboration

The capability of having patients and therapists to remotely interact is important because it can save travel costs, allow more frequent monitoring, and potentially increase access to health care, in particular to remote areas. The motion of each user participating to the virtual collaboration is mapped directly to each respective avatar, and the avatars can be superimposed with transparency or appear side-by-side in the applications. See Figure 6.5 for examples.

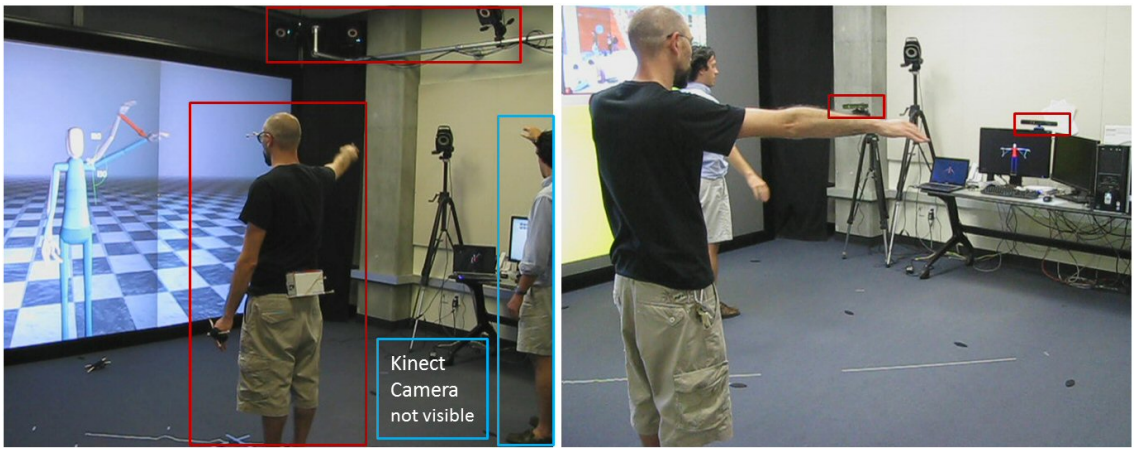


Figure 6.5: Examples of collaborative sessions. Left: one user is being tracked by the high-end system Vicon cameras while the other is being tracked by a Kinect sensor. Right: both users are tracked by Kinect cameras and collaborate with the portable versions of the system running in a desktop and laptop.

The communication between two peers in a collaborative session is based on a client-server UDP communication schema with added packet ordering, guaranteed communication reliability and optional data compression. The server application,

after accepting and validating an incoming connection, starts sending information of the avatar of the current user (sender) and waits for the update of the client's avatar (receiver). For instance, if the therapist application is started as a server, the therapist's avatar becomes the active character in the communication and the second character, the patient's avatar, becomes a receiving entity. If the patient's application is started as the client, the sender entity becomes the character of the patient's application while the tutor/therapist becomes a receiving entity waiting for further updates.

During a networked session each active character maintains a history containing its previous poses and the streamed information between the peers is limited to the information that has changed between the previous frame and the current frame. This feature has been developed to handle communication between peers with limited bandwidth capabilities.

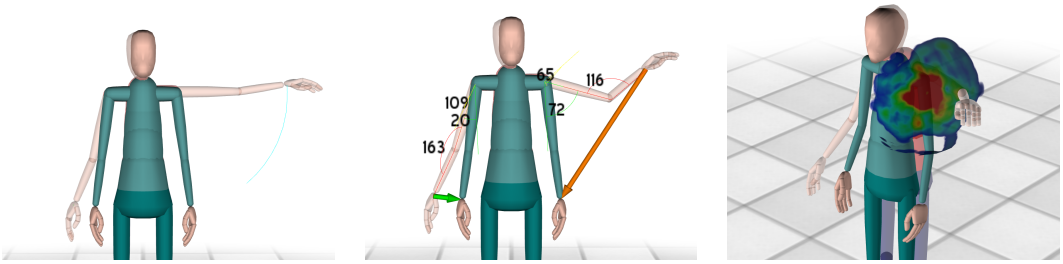
Feedback and analysis tools (described below) are also available during virtual collaboration. The therapist can demonstrate exercises, analyze the patient motion, load preset exercises from the database, watch the patient's performances, record a patient motion in real-time, etc.

6.2.4 Tools for Real-Time Feedback and Post-Analysis

The feedback tools can be activated anytime and they are highly customizable. For example, any joint of the character representation can be tracked and considered for analysis by any tool. Simple commands or text-based configuration files are used for customization. Four types of feedback have been developed in order to provide visual and quantitative information about the user motions in real-time or in post-analysis. The four feedback tools provide information with respect to: trajectories, joint angles, distance to target exercises, and range of motion per exercise. See Figure 6.6.

Trajectories: trajectory trails of selected joints can be displayed in real-time, and they show the performed trajectory of a joint during a fixed past period of time (see Figure 6.6(a)), or after each user's performance, showing the performed trajectory and the trajectory compliance range with the reference exercise. The visualization can be based on fine polygonal segments sampled per frame (for precise analysis for example of tremors), or smoothly generated by B-Spline interpolation.

Angle estimation (virtual goniometer): joint angles can be visualized (Figure 6.6(b)) with a floating label showing the angle value and the local lines used to measure the angle. In practical goniometry for upper-limb physiotherapy [250] angle



(a) Trajectory trails (b) Angles and distances (c) Range of motion

Figure 6.6: Visualization helpers are available for real-time feedback or post-analysis of motions.

measurement is important in order to measure progress and intervention effectiveness, via therapy or via surgery. Therapists can therefore instruct specific movements to patients and observe or log the achieved measurements. The provided angle measurements match the angles measured in practical physiotherapy protocols [250]. The proposed angle measurement is simple and yet flexible to accommodate generic needs.

The angle estimation is calculated as follows: let $p_1, \cdots, p_4 \in \mathbb{R}^3$ be the global positions of the extremities of two dependent (bones sharing a joint) or independent bones, and $R_1, R_2 \in \mathbb{SO}^3$ be the user-defined reference frame rotations. The angle estimation between the limbs at the joint in question is obtained with:

$$\phi = \arccos((R_1 * \|p_2 - p_1\|) \cdot (R_2 * \|p_4 - p_3\|)). \tag{6.1}$$

The proposed method allows the system to measure any kind of angle by just defining pairs of joints and optional reference frame rotations. The tracked angles are specified in the application's configuration file. It gives to the therapist a flexible and easy mechanism to identify and customize the visualization. To isolate angles for upper-arm flexion (extension or abduction) we track, for instance, the angle generated by the scapula/clavicle and humerus, given the scapula bone aligned to the torso as a consequence of the skeleton hierarchical structure. The measured angle is the angle between the arm and the "body line" of the user. In default behavior, angles are only displayed when significant motion is detected. In terms of effectiveness of using Kinect for upper-limb joint angle estimation, the approach has been tested and validated in a similar context [247].

Distances: colored 3D arrows showing the distance between corresponding pairs

of joints, each belonging to a different character, are useful for the patient to track compliance with the demonstrated exercises (see Figure 6.6(b)). The feedback is useful in individual sessions or in remote physical therapy sessions. The distance arrows are employed similarly to the technique proposed by [11]. The arrows are programmed to automatically disappear if the corresponding distance is under a given threshold, and different colors can be associated to different ranges of thresholds. This is in particular useful for slow exercises where compliance is important. Figure 6.7 shows arrow distances enabled together with several angle measurements during execution of one exercise.

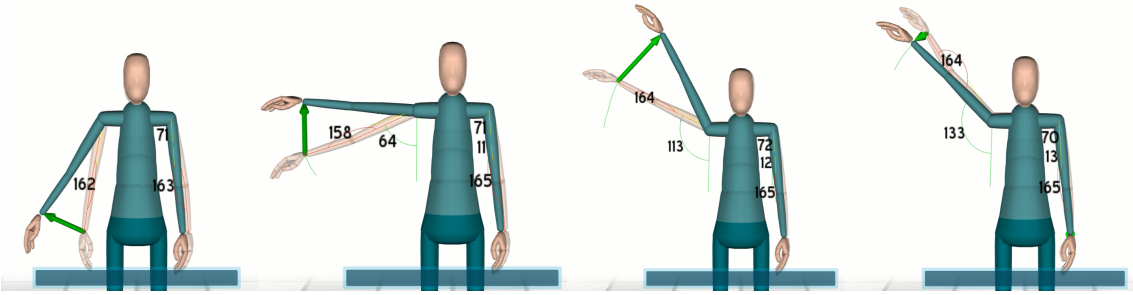


Figure 6.7: Example of several feedback tools enabled while a patient executes a given exercise.

Range of motion: Our range of motion visualization (see Figure 6.6(c)) analyzes the rotation of a selected joint over time. We focus here on the shoulder range of motion evaluation due its importance in rehabilitation of shoulder movements.

The 3 degrees of freedom (DOFs) of the shoulder joint orientation are decomposed into the twist and swing rotation parameterization [105]. The swing motion is then tracked at every frame i, and for each swing orientation s_i measured, the intersection point p_i of the upper-arm skeleton segment at orientation s_i and a sphere centered at the shoulder joint is computed. The history of all traversed p_i points is visualized with colors in the sphere. The sphere is texture-mapped with an image texture initially fully transparent. For every measured point p_i, its position in the texture is determined and the corresponding texture pixel c_i has its color changed. For achieving a clear visualization we employ a relatively high texture resolution and we weight the color increments around c_i with a local Gaussian distribution centered at c_i. The colors are incremented from pure blue to red, providing a colored frequency map of all traversed swing orientations (see Figure 6.6(c)).

The boundary of the colored map will represent the range of motion executed in a given exercise. The original points p_i are also recorded and are used for geometrically

estimating the polygonal boundary describing the full range of motion during a session. This tool provides an excellent way to log improvement of range of motion during rehabilitation, to observe the patient's ability to execute precise trajectories, and to observe if there are areas that are avoided for instance due pain or discomfort. In summary the representation provides a frequency history of the space traversed by the user, and it offers a comprehensive view of the patient's performance. Frequency maps collected per exercise can clearly represent patient progress across therapy sessions.

6.3 Adaptive Exercises

The option of providing customized exercises by demonstration enables the therapist to go beyond recovery plans limited to a set of predefined exercises. The therapist can record his or her demonstrations and then trim, save, load, play, and customize them in different ways. After a validation process the motions can be corrected and/or parameterized. Exercises can then be saved and categorized in a database of exercises. The database is used for fast construction of therapy programs using a desktop-mode interface of the application during consultation with patients.

In order to achieve adaptive exercises we need to address exercise parameterization from the beginning, since the modeling of the exercise motion. Our approach of modeling exercises from demonstration (see Figure 6.8) allows exercises to be generic; however, some structure is expected in order for motion processing algorithms to be able to parameterize the motions in real-time.

Given a captured exercise, we propose correction and parameterization techniques that allow 1) detection and fine-tuning of key characteristics of the exercise such as alignments and constraints, 2) parameterization of the exercise by detecting modifiable properties such as speed, wait times and amplitudes, and 3) real-time motion adaptation by monitoring user performances and updating the exercise parameters in order to improve therapy delivery.

The presented techniques facilitate the process of defining exercises by demonstration by providing several modeling and correction mechanisms and at the same time providing parameterization for real-time adaptation. As a result the proposed methods produce realistic continuous motions that can adapt to user responses in order to improve motivation and outcomes.

Given a new exercise motion demonstrated to the system, the system will analyze the motion and detect the parameterizations that can be employed. An input motion

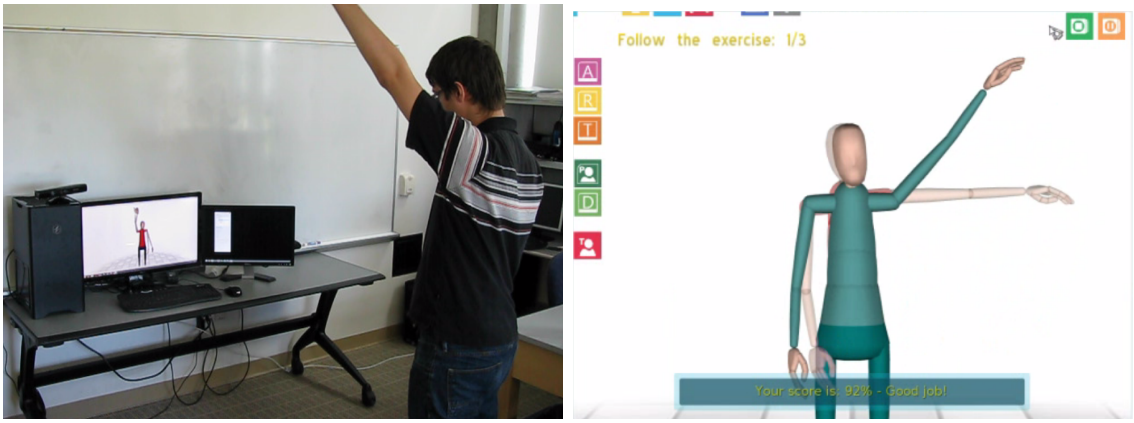


Figure 6.8: Illustration of a modeling session by demonstration using the low-cost Kinect configuration.

is represented as a collection of frames $M_i, i \in \{1, \ldots, n\}$, where each frame M_i is a vector containing the position and the joint angles that define one posture of the character in time.

6.3.1 Detection of Geometrical Constraints

Our constraint detection mechanism is designed for two specific purposes: to help correcting artifacts and noise in the motions, and to provide metrics for quantifying motion compliance. The metrics are used to provide visual feedback to the user, to inform the correctness of performed motions, to make decisions during the real-time adaptation mechanism, and to achieve an overall user performance score for each session.

Appropriate constraints are not constraints which are to be absolutely followed. Recorded motions may have unintended movements and imperfections introduced by the capture system. Constraints must be detected despite these fluctuations, and should be softly enforced so the motion can be made to look correct and also natural.

We analyze the position in space of a specific joint with respect to a frame of reference F which can be placed at any ancestor joint in the skeleton structure. The detection framework can accommodate any desired type of constraint but in this research we focus on two types of constraints: Point and Planar.

The detected constraints are provided to the user and the user then decides 1) if the motion should be modified to better enforce the detected constraint, and 2) if

the constraint is to be monitored during real-time execution of the exercise in order to alert the user every time the constraint is significantly violated. For instance, if the elbow joint is detected to be imovable in an exercise, the system will detect that as a point constraint and may alert the user in real-time everytime the user's elbow is too far away from its point constraint.

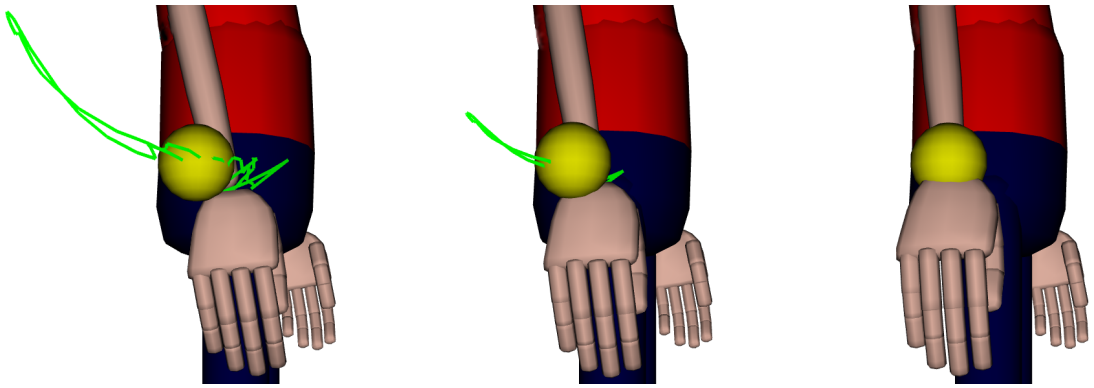


Figure 6.9: Point Constraint. The yellow sphere represents the detection of a point constraint at the elbow joint. From left to right: the wrist motion trajectory (depicted in green) is corrected to the mean point with 0%, 50%, and 100% correction.

• A Point Constraint (Figure 6.9) describes a child joint that is static relative to its parent. Let's $P_i, i \in \{l, \ldots, k\}$ be the cloud of points formed by a joint trajectory with respect to a local frame F generated by re-sampling linearly the motion frames with constant frame rate. The standard deviation of the cloud of points σ is calculated and subsequently checked against a specific threshold α. When the condition is met the current joint is marked as a point constraint and it is represented by the specific point located at the mean μ. When a point constraint is detected the ancestor(s) can be then adjusted to enforce the constraint.

• A Plane Constraint (Figure 6.10) detects if a joint moves approximately within a plane. Similarly to the point constraint detection a point cloud is first generated. Then, PCA is applied to the set of points to determine a proper orthogonal decomposition considering the resulting Eigenspace from the covariance. A planar surface is then retrieved considering the two Eigenvectors with the highest Eigenvalue λ (the magnitude of λ is used to validate the plane). The average distance of the points from this plane is then checked against a threshold β to determine if a plane constraint is appropriate for the given joint.

When a constraint is detected the user is offered a correction percentage to choose. 100% correction results in motion which always tries to obey constraints, whereas

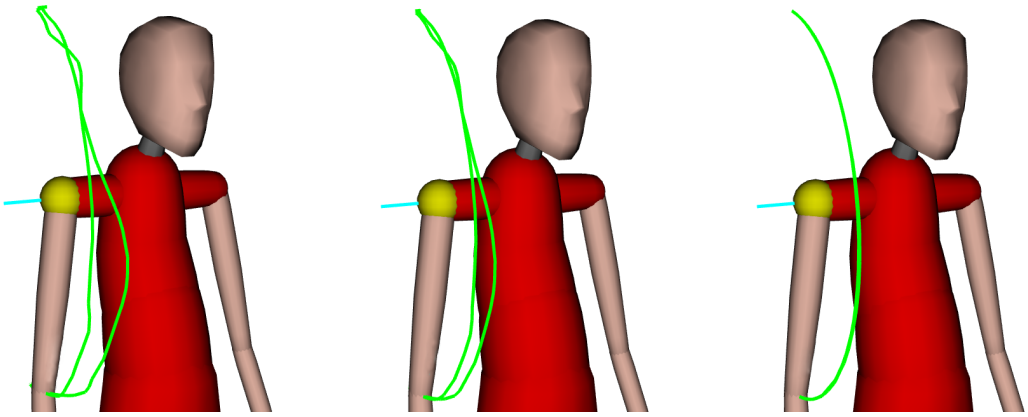


Figure 6.10: Plane Constraint. The blue axis is the normal direction of the detected plane constraint affecting the shoulder joint. From left to right: correction level (from 0% to 100%) where the elbow trajectory (green trajectories) is gradually collapsed into a plane.

0% correction results in no modification the original motion. Figures 6.10 and 6.9 illustrate results obtained for a Plane and Point constraint respectively.

6.3.2 Geometrical Constraint Alignment

Let i be the index of the frame currently evaluated. Let p_i be the position in space of the current joint and q_i be a quaternion representing the current local orientation. A point constraint is defined considering the orientation q_m in the local orientation frame that represents the vector defined by the local point constraint. A point constraint is enforced through spherical linear interpolation between q_i and q_m. Figure 6.9 shows the trajectories generated by the wrist joint collapsing into a point constraint.

To apply the plane constraint, we identify the orientation defined by the projection of each point p_i to the plane discovered during the detection phase. The plane constraint is then enforced, similarly to the point constraint, by interpolating the equivalent orientations. Figure 6.10 shows the trajectories generated by the elbow joint aligning into a plane constraint.

6.3.3 Detection of Exercise Parameterization

Consider a typical shoulder flexion exercise where the arm is raised until it reaches the vertical position or more (initial phase); subsequently the arm is held for a few seconds (hold phase) and then it relaxes back to a rest position (return phase). This is the type of exercise that we seek to parameterize.

The analysis procedure makes the following assumptions: a) each motion M represents one cycle of a cyclic arm exercise; b) the first frame of a motion contains a posture representing the starting point of the exercise; c) the exercise will have distinct phases: the initial phase (M_{init}) is when the arm moves from the initial posture towards a posture of maximum exercise amplitude, then the exercise may or not have a hold phase (M_{hold}) but at some point the exercise must enter the return phase (M_{end}), where the exercise returns to a posture similar to the starting posture. In addition, if the motion contains a hold phase at the point of maximum amplitude, it will mean that an approximately static pose of some duration (the hold phase duration) exists at the maximum amplitude point. We also consider an optional 4^{th} phase that can be added to any exercise, the wait phase (M_{wait}), which is an optional period of time where the character just waits in its rest pose before performing a new repetition of the exercise. Figure 6.11 illustrates a typical exercise that fits our assumptions.

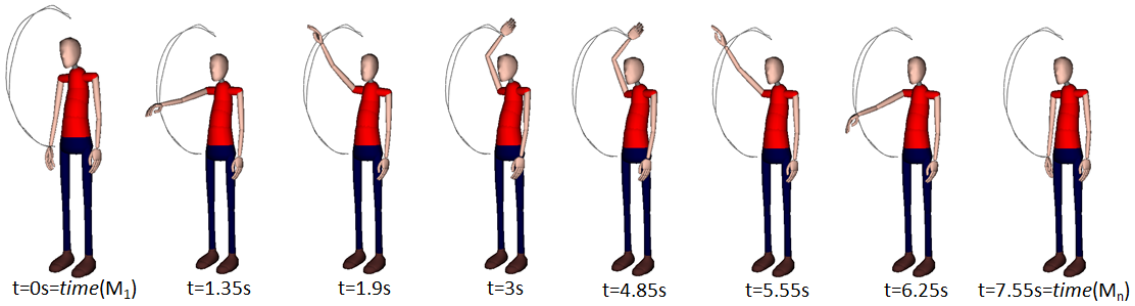


Figure 6.11: Example of a typical exercise captured from a therapist in one of our tests with the system. The shown trajectory is the trajectory of the right wrist joint along the entire motion. The initial phase happens between t=0s and t=3s. Then, between t=3s and t=4.85s there is a hold phase at maximum amplitude where the therapist is static (but small posture variations are always noticeable). Then, between t=4.85s and t=7.55s we can observe the return phase.

The analysis of the exercise to be parameterized has two main steps: first the arm to be parameterized is detected; and then the two *motion apices* are detected. The

apices, or the points of maximum amplitude, are the intersection points between the initial and return phases with the hold phase (frames $t = 3s$ and $t = 4.85$ in Figure 6.11). These points will be a single apex point if the motion has no hold phase in it. If the phases above are executed successfully the input motion is segmented in initial, return and an optional hold phase, and the motion can be parameterized.

In order to detect which arm to parameterize we extract the global positions of the left and right wrists along their trajectories. Let L_i and R_i respectively denote these positions. Since our focus is on arm exercises the wrist represents an obvious distal joint of the arm kinematic chain to use in our parameterization analysis algorithm. For each wrist trajectory L and R we compute the 3D bounding box of the 3D trajectory. The bounding box dimension is used to determine which arm is moving and if the motion can be parameterized. As a result of this process, the analysis will return one of the following four options: a) the motion cannot be parameterized; b) the motion will be parameterized by the left/right arm; or d) the motion will be parameterized by both arms (targeting symmetrical exercises).

6.3.4 Exercise Parameterization

If the motion can be parameterized and its type is determined, we then search the motion for the points of maximum amplitude. To detect one apex point we search for a frame that indicates a sharp turn in trajectory. Since the motion may or not contain a hold phase, we perform the search in two steps: a forward search starting from M_1, and a backward search starting from M_n.

Let i be the index of the current frame being evaluated (M_i). Let T represent the trajectory of the left or right wrist joint, that is, T_i will be R_i or L_i (the trajectory is first smoothed through moving mean to reduce sensor noise). In order to determine if M_i represents an apex point we perform the following steps. We discard the initial points until the distance between two consecutive points becomes greater than a specific threshold d_t (a threshold of 5cm worked well in practice). We first compute the incoming and outgoing direction vectors with respect to T_i, respectively: $a = T_i - T_{i-1}$, and $b = T_{i+1} - T_i$. If a or b is a null vector, that means we are in a stationary pose and we therefore skip frame M_i and no apex is detected at position i. Otherwise, the angle α between vectors a and b is computed and used to determine if there is a sharp change in direction at position i. If α is greater than a threshold angle, frame i is considered a probable apex point, otherwise we skip and proceed with the search. We are using a threshold of 75 degrees and this value has worked well in all our examples with clear detections achieved. To mark an apex to be

definitive we consider the distance between the following k frames to be less than d_t.

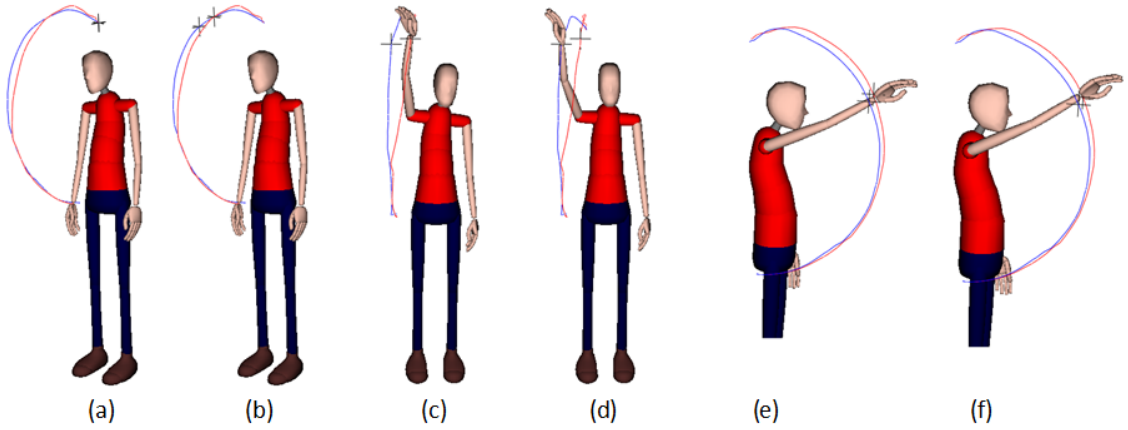


Figure 6.12: The red trajectory shows the initial phase M_{init}. The blue trajectory shows the return phase M_{ret}. The input motion is the same as Figure 6.11. (a) The full (100%) amplitude of the input motion is shown by the trajectories. Two black crosses at the end of the trajectories (in almost identical positions) mark the positions of Apex 1 and Apex 2. (b) The two black crosses now mark the maximum amplitude points in the initial and return trajectories at 75% amplitude. (c,d) In this frontal view it is possible to notice that the postures at 75% amplitude in the initial and return phases are different. The hold phase will start by holding the posture shown in (c), and when the hold phase is over, we blend into the return motion at the posture shown in (d) in order to produce a smooth transition into the return phase. (e,f) Lateral view.

The test described above is first employed for finding the first apex point by searching forward all frames (starting from the first frame). The first apex found is called Apex 1 and its frame index is denoted as a_1. If no apex is found the motion cannot be parameterized. If Apex 1 is successfully found, then the search is employed backwards starting from the last frame, however not allowing passing beyond Apex 1. The second apex found is called Apex 2 (a_2). Note that Apex 2 may be the same as Apex 1, in which case no holding phase is present in the input motion. After the described analysis, the main three portions of the motion have been detected: a) the initial phase is defined by frames $\{1, 2, \ldots, a_1\}$ (motion segment M_{init}); b) the hold phase is defined by frames $\{a_1, a_1 + 1, \ldots, a_2\}$, if $a_2 > a_1$, and nonexistent otherwise; and c) the return phase is defined by frames $\{a_2, a_2 + 1, \ldots, n\}$ (motion segment M_{ret}). Once an input motion M is successfully segmented, it can then be parameterized.

Amplitude and Hold Phase Parameterization We parameterize amplitude in terms of a percentage of the wrist trajectory: 100% means that the full amplitude observed in the input motion M is to be preserved, if 80% is given then the produced parameterized motion should go into hold or return phase when 80% of the original amplitude is reached, and so on. Let h be the time duration in seconds of the desired hold duration. When the target amplitude is reached, the posture at the target amplitude is maintained for the given duration h of the desired hold phase. When the hold phase ends, the posture is blended into the return motion M_{ret} at the current amplitude point towards the final frame of M_{ret}. See Figure 6.12. The described operations are enough to achieve a continuous parameterized motion, however two undesired effects may happen: a noticeable abrupt stop of M_{init} or an unnatural start of M_{ret}, because the parameterization may suddenly blend motions to transition at points with some significant velocity. To correct this we re-time the segments so that motion phases always exhibit ease-in or ease-out profiles.

Behavior During Hold and Wait Phases In order to improve the realism, we add a small oscillatory spine movement mimicking a breathing motion, which is applied to spine joints during the hold and wait phases. One particular problem that is addressed here is to produce an oscillatory motion that ends with no contribution to the original pose at the end of the oscillation period. This is needed so that, after the oscillation period, the motion can naturally continue towards its next phase and without additional blending operations. We thus have to produce oscillations of controlled amplitude and period. This is accomplished with the following function: $f(t) = d\sin(t\pi/d)$, if $d < 1$, and $\sin(t\pi/(d/floor(d)))$ otherwise; where $d > 0$ is the duration of the oscillation period, which in our case will be the duration of the hold or wait periods.

At the beginning of a hold or wait phase we save the joint angles of the spine in a vector s, and then apply to the spine joints the values of $s + cf(t)$, where $t \in [0, d]$, and c is an amplitude constant. We obtained good behavior with $c = 0.007$, and only operating on one degree of freedom of two spine joints: one near the root of the character hierarchy, and one about the center of the torso. The used degree of freedom is the one that produces rotations on the sagittal plane of the character.

Overall Parameterization The described procedures allow us to parameterize an input motion M with respect to up to four parameters: amplitude a (in percentage), hold time h (in seconds), wait time w (in seconds), and speed s (as a multiplier to the original time parameterization). Given a set of parameters (a, h, w, s), the input motion can be prepared for parameterization very efficiently and then, during execution of the parameterized motion, only trivial blending operations are

performed in real-time.

6.3.5 Real-Time Adaptation

When the adaptation mechanism is enabled the system collects information from the patient's performance in real-time and adapts the current exercise in its next repetition. In addition, visual feedback is also provided: arrows showing direction of correction for improving motion compliance, constraint violation feedback and also an overall performance score with explanatory text (see Figure 6.13).

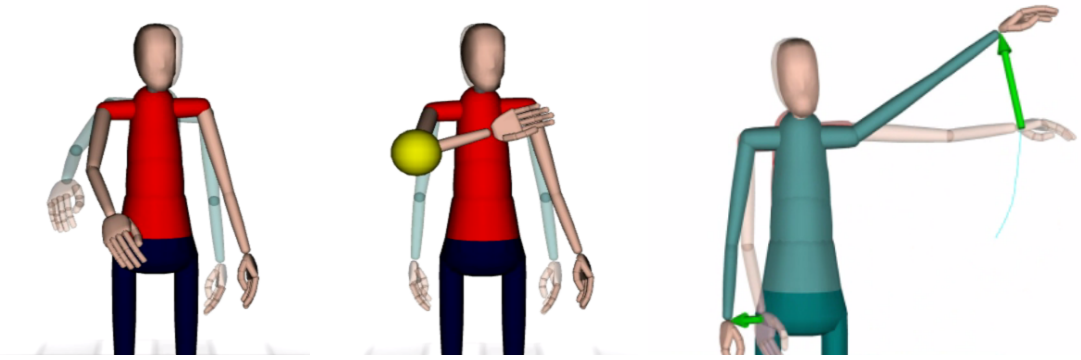


Figure 6.13: The red character displays the user's motion and the blue one the target exercise. Left: no violated constraints. Center: user is reminded to correct the elbow. Right: arrows show direction of correction to improve compliance.

Four types of adaptation mechanisms are provided:

• **Amplitude Adaptation** The range can vary from 75% to 100% of the target amplitude. The system tracks the distance between the patient's active end-effector and the apex at the target amplitude position. If the minimum distance is larger than the amplitude compliance parameter specified by the therapist, the next exercise execution will have the target amplitude lowered to become within the compliance range. If in a subsequent repetition the user reaches the current (reduced) target amplitude, then the next target amplitude will be increased towards the original target amplitude.

• **Hold Time** The hold phase adaptation is designed to adapt the time at hold stance to improve resistance, usually in a posture that becomes difficult to maintain over time. The maximum distance between the target hold point and the performed

end-effector position is computed. If above a threshold, the patient is having difficulty in maintaining the posture and the next exercise repetition will have a shorter hold phase duration. If in a subsequent repetition the patient is able to well maintain the hold posture, then the hold duration is gradually increased back to its previous value.

• **Speed Execution** During patient monitoring, the active position of the patient's end-effector is tracked and its distance to the demonstrated exercise end-effector is computed for every frame. If the average distance computed across the entire exercise is above a given trajecory compliance threshold (see Figure 6.14), the next exercise execution speed is decreased. If in a subsequent repetition the difference is under the threshold the speed will be gradually adjusted back.

• **Wait-Time Between Exercises** The initial wait time specified by the therapist is decreased or increased in order to allow the patient to have an appropriate time to rest between exercises. A performance metric based on averaging the trajectory compliance and the hold phase completion metrics is used to determine how well the patient is being able to follow an exercise. If the user is performing the exercises well a shorter wait time is selected, otherwise a longer wait time is preferred. In this way wait times are related to the experienced difficulty in each exercise, and they adapt to specific individuals and progress rates.

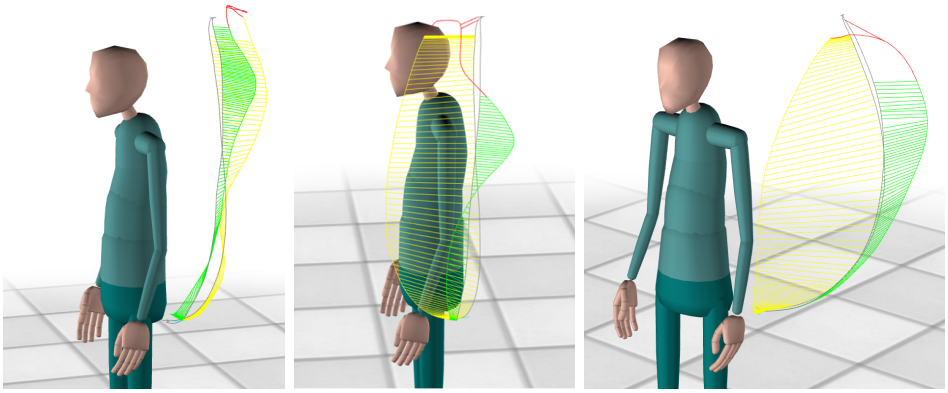


Figure 6.14: From left to right: high, medium and low trajectory compliance.

The described adaptation mechanisms have been identified as a first set of relevant strategies after many discussions and interactions with therapists. In the next section we present a summary of the main feedback received.

6.4 Feedback, Results and Discussion

Since the beginning of the development of our system we have worked closely with therapists at the University of California Davis Medical Center to validate the complete therapy creation, delivery and adaptation pipeline. This joint collaboration was also critical to identify the described ways to delivery and adapt the therapy plan to patients. We gathered feedback in two phases. In the first phase, focus groups were held and open ended questions elicited multiple responses of major factors that should be considered in exercise prescription. In the second phase we demonstrated the current prototype application and then asked the therapists for their feedback.

In the first phase questionnaires were distributed to 40 staff therapists asking about the importance of individualized interactions, the factors used to determine correctness of performed exercises, and the motivational and adaptation strategies commonly used by therapists. For each question, the therapists were asked to rank the factors identified in the first phase between 1 (not important) and 5 (highest importance). The factors that were ranked 4 or 5 by more than 20 therapists are summarized in Figure 6.15 for selected questions.

From the collected data summarized in Figure 6.15 it is possible to make several observations. Performing exercises in a correct manner is largely related to being close to the prescribed exercises, what is translated in terms of not having compensatory movements and maintaining correct postures and trajectories. The visual feedback tools that were described well address these issues. In addition, the proposed constraint detection methods for real-time warning if the user performs motions that do not respect constraints also well address enforcing correct execution of exercises.

One point that cannot be addressed by therapy systems that only give visual output is to provide tactile feedback. However we point out that tactile feedback was considered as important as visual feedback, which is well addressed by our system. The fact that visual and audio feedback were highly ranked is also important because it indicates that they may well compensate for the lack of tactile feedback, which is at the same time a desirable characteristic of the system from a safety perspective.

Several causes were cited as reasons justifying the need for exercise adaptation, for example, the patient's ability to learn, patient improvement, decreased pain, increased strength, etc. The proposed adjustment of wait and hold times, exercise speed and amplitude provide direct ways to adapt the exercise as the patient progresses. In particular, it is also important to adapt in a constant basis given the patient's ability to learn the exercises. At the beginning of an exercise set it is of-

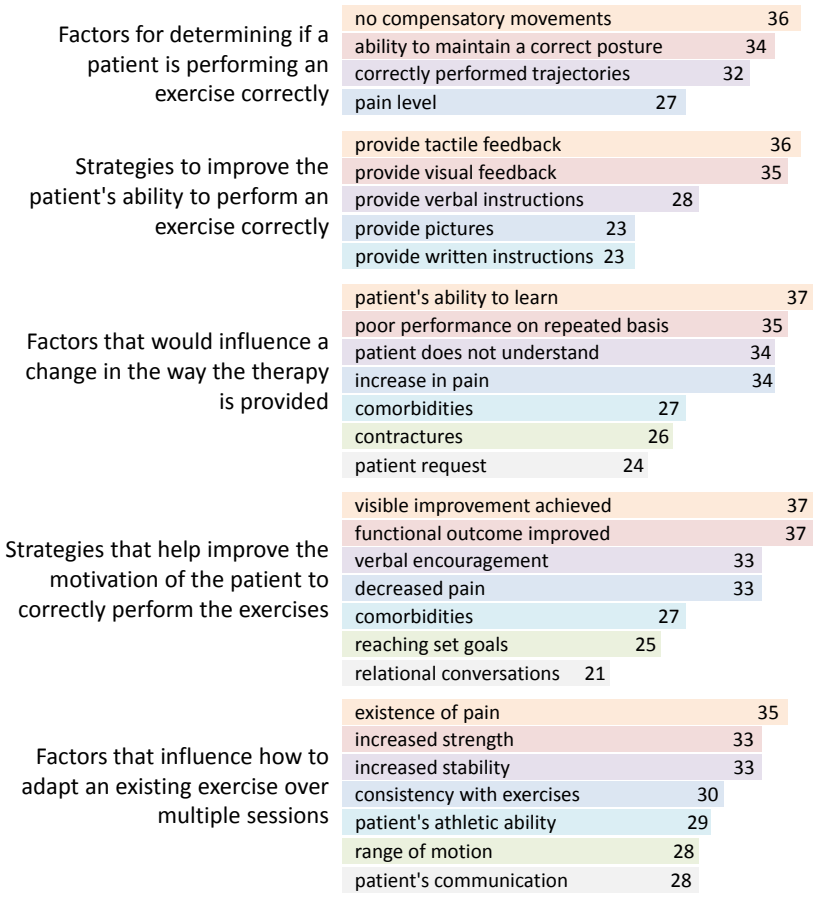


Figure 6.15: Summarized responses. The numbers show how many therapists (out of 40) rated each factor as 4 or 5.

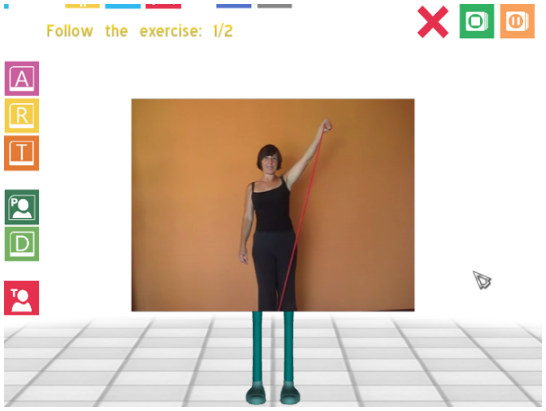
ten observed that patients need more time to assimilate an exercise while at the end of a set the patients are well able to perform them quicker and with less wait times. The same can be observed in subsequent sessions, however usually with progressively faster learning rates. The proposed adaptation methods are capable to adjust to patients as needed, thus significantly improving correct exercise execution, improvement observation, engagement, and adherence to the therapy program.

In the second phase we asked questions to the participants before and after they saw the capabilities of our system. The questionnaire consisted of generic questions as follow up of the first phase questionnaire as well open-ended suggestions and preferences to improve the current setup. At first only 45% of the participants were

confident that patients do exercises consistently and correctly at home, but after seeing our system (and if our system was to be employed) that percentage raised to 70%. When asked about the importance of modifying exercises during the progress of the therapy, 70% cited as very important and after seeing our system this percentage was even raised to 85%.

These results indicate that adaptation is an important factor for achieving VR-based therapy systems that have real potential to be adopted. While the presented solutions provide only first results towards addressing adaptation strategies, we hope that the described framework and adaptation techniques provide a significant step towards the right direction.

(a) Interface for patients to start a single exercise or the daily recovery plan



(b) Example of exercise demonstrated to the patient through video

(c) Interface to generate therapy plans for patients with the possibility to enable the exercise parameterization

(d) Parameterization interface of a single exercise with motion preview

Figure 6.16: Examples of the user interface for therapists (sub-figures [c,d]) and patients (sub-figures [a,b]) in the low-cost solution for in-clinics setup.

CHAPTER 7

Final Conclusions

Realistic and interactive virtual humans designed to deliver motion based information in training systems are challenging to create. Nevertheless, virtual humans have great potential to become as effective as human trainers in monitored, feedback-based, virtual environments for training and learning where human-oriented motion skills or human-conveyed information are key to the learning material. In this final chapter I discuss some of the key points of my work with concluding remarks.

In this dissertation, I have presented new approaches to develop virtual human training systems based on direct demonstrations. The system achieves an effective overall design for modeling and demonstrating parameterized motions; combining realism, flexibility and adaptability for several training scenarios. Additionally, I have delineated several guidelines for the employment of avatars in immersive training and rehabilitation systems and I have applied these findings to the concrete scenario of delivering upper-body physical therapy.

While the ultimate goal of building the perfect Virtual Human designed for training still needs to be brought to life, significant achievements toward this goal are presented by this dissertation.

7.1 Summary of Contributions

The most important results of this dissertation can be largely summarized considering as described in the following paragraphs.

First, we introduced an immersive motion modeling platform based on the direct demonstration and parameterization of motions where experts in a specific field can model actions and gestures by demonstration, without the need of having previous experience with the system, and directly inside a virtual environment. The interface proposed allows to interactively build clusters of example motions in real-time by demonstration. After demonstrating the needed motions, the virtual trainer is then able to synthesize motions that are similar to the provided examples and at the

same time that are parameterized to generic targets and constraints, and they can also be reproduced in environments different from where they where recorded. The proposed approach is based on blending a set of consistent time-aligned example motion sequences and the blending weights are directly optimized until the specified constraints are best met (this process in general takes a few milliseconds depending on the dataset recorded). This reproduction phase can be adopted for training of non-expert subjects.

Second, I performed several experiments, collecting motion data and user experience, to analyze the effectiveness of virtual trainers with respect to different choices on display size, use of user perspective stereo vision, and the use of avatars, in which case the task is transferred to the avatar space (with the avatar mimicking the motions of the user). The systematic study investigated the use of user-perspective stereo vision both in direct interaction mode and in 3rd-person mode with the avatar being driven by the user to perform a given task.

Several conclusions can be drawn by this study but in summary, if the involved tasks require generation or reproduction of motions with desired qualities, such as in training applications where the environment or objects and tools are key factors, the use of avatars and stereo vision will improve that ability. However, if the goal is to accomplish tasks no matter the type of motions used, direct interaction in user-perspective will be more efficient.

In addition other important remarks can be highlighted:

- User-perspective stereo vision with direct interaction is the optimal choice in terms of task accuracy and completion time during precision tasks (users were 6 times more accurate and 2 times faster); when the task was transferred to the avatar's space users were 3 times more accurate.
- User-perspective stereo vision, when the task involved motion reproduction, showed users' improvements both in terms of synchronization and compliance with the reference motion.
- The use of avatars produced increased attention to the avatar space, allowing users to better observe and address motion constraints and qualities.
- Coupling avatar use with stereo vision resulted in users paying more attention to the motions within the virtual environment, improving the realism and correctness of the motions. These findings represent key factors to consider when designing applications for distant training in collaborative spaces where the kinesthetic component is fundamental with respect to the environment.

- The use of avatars was shown to improve motion reproduction in the cases where stereo vision was not present. This factor suggests that in applications where stereo vision is not practical to be used (such as in homes or clinics), the use of avatars can improve the understanding of motions displayed by virtual tutors.

- The use of small displays induced users to not overestimate their capabilities during precision tasks however, increasing their execution time and frustration.

We have also observed that there are critical design choices that highly influence the suitability of the system to different types of interaction needs. For example, during the motion modeling phase of our direct demonstration framework, users will benefit from the combined adoption of user-perspective stereo vision and avatar if the task involves modeling of gestures with respect to a given scenario or during object manipulation. Moreover, trainees that have to reproduce motions during a demonstration phase benefit from the presence of an avatar mimicking their motion.

Finally, we applied the previously described concepts and guidelines to develop applications and tools to the practical scenario of delivering monitored upper-body physical therapy. New methods for exercise modeling, parameterization, and adaptation are presented in order to allow therapists to intuitively create, edit and re-use customized exercise programs that are responsive and adaptive to the needs of their patients. Exercises can be configured to particular patients, both in terms of creating new exercises as needed and in terms of designing how exercises should adapt to patient preferences, physical limitations, and recovery rates. The proposed solutions were evaluated by therapists and demonstrate the suitability of the approach.

In addition to the main contributions, in order to develop these collaborative and immersive virtual training systems with full-body avatar interaction and direct demonstration we introduced a new development middleware based on a popular and advanced open-source game engine. The infrastructure is scalable to different hardware configurations and defines high-level programming abstractions to hide, simplify and automate several aspects of the creation of distributed virtual reality applications. The system has been tested and compared with other analogous solutions demonstrating similar performances without the use of customized hardware.

In summary, this dissertation contributes to both substantive as well as methodological questions related to the evaluation and the adoption of virtual humans in training scenarios using the direct demonstration approach.

7.2 Directions for Future Work

The immersive motion modeling techniques and applications presented in this dissertation demonstrated the successful overcoming of several challenges related to the development of virtual human training systems based on direct demonstrations. Future research directions can be identified from the findings and remarks provided by this dissertation. We envision several possible directions for extending the presented work.

The approach of modeling motions by direct demonstration has great potential to be effectively employed in a number of applications; however, there is room for future research in all the aspects involved in the process. Considering the user's motion reconstruction from reduced marker set (or more generically from any motion reconstruction sensor connected to the system) motions should be validated against a set of measurements that objectively evaluates the degree of reliability on each tracked body part.

Several approaches have been explored using a database of motions to assist with motion reconstruction by querying a latent space built from the motions in the database. The possibility of using the user's motion as an input to the optimization search in the latent can provide a high-quality way to synthesize complete postures from the incomplete input user posture. The challenge to be addressed is to achieve real-time performances for use in interactive systems and also optimizing and refining the coverage of the space.

In addition to this, the integration in the motion modeling phase of learning policies will be necessary in order to categorize and parameterize motions not only in terms of motion synthesis but also in terms of concatenation and sequences (where actions are composed by sequences of motions and gestures strictly connected by relations). The imitation problem can then be restated as the multi-modal imitation of complex motor actions that involves learning a sequence of primitives and interconnection probabilities. Having a complete policy and a graph connection structure would also be beneficial for modeling of complex virtual trainers' behaviors (such as speech correlated with gestures) that are, at the present, modeled manually by animators.

Another promising and current research direction is based on the integration of dynamic models in the motion demonstration phase. The integration of physically simulated stimuli to the employed motion generation algorithms would produce virtual trainers that are responsive to the environment (due collisions and gravity forces) and would as well generate motions dynamically feasible when interacting with or

holding objects. For example, this would be useful in the case of generating virtual exercises coupled with physical therapy tools such as elastic bands for resistance training. Additional examples include complementing the motion synthesis in scenarios with limited motion capture examples.

On the same note, another important extension that will lead to several new research opportunities is the generation of parameterized motions of a virtual trainer (feedback, environment or tools/objects states) in response to the actions performed by a user's motions in real-time. For example, consider the complex scenario of training medical practices (e.g. CPR); in this case more than one virtual trainer can be involved introducing the new challenge of synthesizing motions with orchestration. The goal would still be to maintain the fidelity of the original motions and at the same time coordinating and parameterizing gestures in accordance with the trainee performance. In addition, another example can be seen considering the case of training students on archeology excavation practices. The motion produced by the virtual trainer should be adapted to the changes made to the environment in accordance to specific gestures, actions or important gestures' strokes. A popular approach to address these challenges in Computer Animation is the adoption of Interaction Meshes [122]. While this method works well in practice it requires action and feedback modeling for both virtual trainer and trainee before hand. In this case the extensibility of the application would be compromised since the modeling phase would require modeling both actions and behaviors going against the assumption of modeling only expert users' motions by demonstration.

Even though our approach to immersive and home-based VR for physical therapy efficiently improves many aspects in the process of delivering ad-hoc physical therapy several key questions remain to be addressed if the promise of mass distributed VR physical therapy is to become a reality. First, we need to better understand more in depth if and in which conditions VR and avatar-based rehabilitation systems improve the patient adherence to the therapy and ultimately his or her recovery. Due to the importance and difficulty of evaluating the quality of the result and the effectiveness of the applied techniques in real training and rehabilitation scenarios, a consistent and systematic evaluation and analysis of the overall systems should be performed, considering the overall recovery period as well with patients engaging in normal procedures during standard physical therapy. The evaluation procedure and trials should also not only consider the viability of the technologies applied but also the patient outcomes. Fundamental questions regarding the provision of therapy can also be investigated employing standardized technology-based rehabilitation practices, for example timing of therapy, impact of lesion location, type of learning (e.g. implicit vs. explicit), feedback type and schedule, and so on.

Second, the human-computer interface as well as the exercise delivery and acquisition components are also important aspects that would lead to novel research, where methods need to be developed to ensure that VR applications work as well at home as in laboratory settings. The adoption of a more precise, non-invasive and stable tracking system will enable complex physical therapy scenarios where objects would not interfere with the therapy delivery system (e.g. steps, bars, weights, chairs and so on). New displays such as auto-stereoscopic displays (no glasses needed) are likely to provide a more user-friendly component that is much less prone to cause problems or to disrupt home-based training or in clinics maintenance. There is also a clear need for studies to help develop protocols for managing and deploying the overall therapy solution considering dierent populations in terms of disabilities, ethnicity, age, personal preferences, etc.

Finally, another key factor for delivering effective physical therapy consists on maintaining the user's expectation high during the overall recovery period. This fact could be addressed by introducing in the common VR based pipeline social methodologies employed in popular collaborative video games (mmorpgs, social network games and freemium services) such as shared rewards, leader boards and winning policies. The social components in games are the most prominent and important factor nowadays that make a game successful and defines its longevity. For example, introducing physical therapy tasks to retrieve points that can then subsequently spent in a common game mechanism with storyline and shared interaction mechanisms (e.g. Farmville or similar titles) that would increase users' expectations and motivation. This reward based system should be publicly connected in order to also allow post-therapy community interaction with patients and clinicians (chats, rewards exchange and public in-game posts and achievements).

APPENDIX A

3D Printing and Immersive Visualization for Improved Perception and Interaction with Past Material Culture

The work presented in this appendix is not directly related to the topic of avatar-based training systems; however, it involves perception studies relevant to the topic of this dissertation. It was developed in collaboration with researchers in archeology [78].

Object manipulation is an important element in understanding and interpreting past material culture. Tactile perception of physical qualities is important for feeling, interpreting, and understanding ancient artifacts. However, sight is often given priority over the other senses when people experience with such kind of objects. Visitors of archaeological sites and museums are usually not allowed to touch archaeological remains for obvious reasons of conservation and preservation. Curatorial restrictions are intrinsic to ancient artifacts; however, they deprive visitors of "the possibilities to grasp the objects' material, sensorially perceptible characteristics, which are pre-existing and inherent, real and physical" [85].

In order to overcome the limitations related to the inability of handling objects in museums and archaeological areas, 3D technologies have been employed to provide new ways to experience with our material past. Significant recent efforts in this area have been made to well reproduce sensorial experiences with past material culture. Immersive virtual reality systems are one of the ways in which people can grasp the materiality of the past by interacting with virtual reproductions of artifacts. Even if tactile feedback is not present, virtual manipulation experiences are rich and the approach has been increasingly used in museums and research labs. In addition, museums and research facilities have recognized the value of 3D printing for research and for the presentation of artifacts to the general public. These new ways of presentation enhance multiple sensorial experiences with our past, and present new research questions on how people negotiate with the inauthentic.

In order to correctly explore the benefit of these new technologies, it is important

to understand how experiences with 3D digital copies in a virtual environment and with real 3D prints differ from the usual visual experience people have with original artifacts preserved and displayed inside museums. We present in this chapter three experiments designed to investigate these points. We are particularly interested in how people interact with 3D digital copies of artifacts, 3D prints and digital reconstructions in an immersive stereoscopic system, and how these experiences differs from the visual experience with original artifacts and with tactile experiences with 3D prints. Even though many studies in computer and cognitive sciences have explored how people perceive specific characteristics of objects (e.g., weight, size, density etc.) through visual, tactile, and virtual experiences, little is known about how people perceive past material culture through the senses, and how experiencing ancient artifacts through different media affects the perception of our past. Through a set of experiments designed to investigate how people respond to 3D virtual and printed replicas of artifacts, this research addresses perception of artifacts with the goal of identifying improved experiences for displays in museums.

The remainder of the chapter is organized as follows. Section A.1 discusses related work. Section A.2 presents the first experiment, which investigates how people perceive physical characteristics of ancient artifacts and how different media affect this perception. Section A.3 describes the second experiment, which investigates how people describe artifacts through bodily movements and how different media affect the production of gestures. This experiment evaluates the concept that gestures and words are both part of a thinking process [220]. The analysis of gestures therefore helps to understand how people think and engage with artifacts and their virtual and 3D printed counterparts. Section A.4 presents the third experiment, which investigates how people engage with artifacts in different media states. This experiment was designed to collect metacognitive information on how participants considered each experience to be useful for the perception and understanding of the artifacts, and how engaging the experience was perceived to be in each condition. Finally, Section A.5 discusses major findings and Section A.6 concludes the chapter and proposes future research.

A.1 Background and Related Work

A.1.1 Studies on how we think with artifacts

Scholars in psychology and cognitive sciences argue that when people engage with material objects, they think with them [68, 138, 179, 251, 279, 333, 359, 372]. To

explore how people use objects as vehicle of thought, Kirsh [161, 163] used the example of a 6-piece puzzle. In a physical condition, people can move these six pieces and physically try to assemble them and create an image. In a mental imagery condition (i.e., when people cannot touch the pieces), people virtually move these pieces in their head (i.e., mental rotation and assembly). Both activities (i.e., the physical and the mental) show how our thoughts include material objects [163]. When we think through external representations, we can compare objects, build on them, rearrange them (as shown by the example of the puzzle), recast them, and perform other kinds of manipulations. Through these activities we are able to deepen our understanding of objects. According to Kirsh [163] however, all these arguments focus on material vehicles that represent propositional thought (i.e., abstract logic), and artifacts may mediate thought differently. They may have more to do with non-linguistic thinking. The question here is: "How do people co-opt non-propositional objects for thought?" ([163]; emphasis original). In other words, how do people engage with material objects?

Tactile perception of a real-life object is usually an active experience involving information gathered from a variety of senses related to touch, such as texture and temperature, as well as movement and position of the hands and fingers during identification [98]. Touch provides an understanding of shape, size, and weight, and it is through this sense that people develop an understanding of other properties such as density, and all key properties for the exploration of artifacts [84, 162]. For example, assessing the weight of an object can be critical for determining its function. Through several experiments Klatzky and colleagues have shown that people are relatively competent at recognizing objects haptically (i.e., through the sense of touch). In one experiment Klatzky [166] asked blindfolded people to recognize common objects just by touching them, and these people did so with very few inaccuracies. Subsequent studies clarified how people haptically explore objects to recognize them. These studies show how people actively explore their environment, executing a series of specific classes of hand movements in search of the perceptual attributes (i.e., texture, size, weight, etc.) of objects [189]. However, similar studies have shown the perception of certain characteristics is not merely a haptic phenomenon. Some experiments have shown, for instance, that when two equally heavy objects of different sizes are lifted, the smaller object is perceived as being heavier (size-weight illusion; [119]). This finding demonstrates a visual bias affecting the perception of artifacts. Heineken et al. [119] have also shown that an object's weight estimation is affected by the medium selected to present the object (e.g., 3D digital reproduction vs tactile experience with original objects) and that the more presence is experienced in a computer generated environment, the more realistic dig-

ital objects appear. A complete digression on tactile and haptic illusion can be found in the survey proposed by Lederman et al. [188].

Tactile experience is also considered an effective means to interpret ancient artifacts. MacGregor [210] suggested that a haptic analysis of material culture is an avenue available to the archaeological interpretation of past sensory orders, and that this analysis is conceptually and functionally different from analyses made using static visual images. For instance, when scholars studied carved stone balls circulating in the Aberdeenshire region of Scotland during the third and second millennia BC (1852-55 BC) they frequently made reference to their appearance (decoration and number of knobs) in support of the interpretation that these balls were used in a ceremonial context to enhance the social status of those holding them. Clearly, scholars privileged vision above all other senses. According to MacGregor, however, when someone holds a carved stone ball decorated with knobs and rotates it quickly, the object visually takes another form, becoming a complete sphere (i.e., the knobs visually disappear). This transformation of the objects could have been witnessed by a much larger group of people and may have been considered magical. In this case, the haptic analysis of the balls results in a new interpretation of the object function.

A.1.2 The use of technologies for improving the museum experience: haptic interfaces, augmented reality, virtual reality, and rapid prototyping techniques

The studies discussed above show how important it is to manipulate objects in order to activate thinking processes that help with the interpretation of past material culture. To respond to this need of physical manipulation, computer scientists have sought to develop complex systems that simulate the tactile experience with real-life objects. Over more than twenty years, they have designed devices able to reproduce the feel of physical contact with objects and the perception of tactile stimuli (i.e., *haptic interfaces* and *force-feedback*). Haptic Interfaces (from now on HI) and force-feedback have been widely studied in the last 20 years [50, 107, 147], and have been commercialized by companies such as Sensable and Immersion. Haptic systems have been designed for experimenting with texture feeling [227, 70] or with weight feeling integrated in immersive virtual environments [135]. A few studies show how HI can be applied to create virtual art and archaeology exhibitions wherein users interact with both the visual and haptic senses [46, 32, 340, 214, 205, 375].

Although many projects in computer science have been concerned with reproducing real-life tactile experiences with material culture, these projects do not yet

allow a widespread use of HI for 3D museum and research applications in heritage and archaeology.

Nonetheless, museums are keen on presenting their collections through the use of new technologies, to attract diverse audiences. Another key element to fill the gap between real and digital is augmented reality. Augmented reality (AR) is a real-time view of real-world environments augmented by computer-generated sensory input such as sound, video, and graphics. Augmented reality, unlike virtual reality (VR), tries to enrich reality instead of just reproducing it [156, 211, 31]. As a result, the technology enhances one's current perception of reality. The effects of immersive virtual reality on scientific visualization, data analysis and in human interaction tasks have been studied extensively (for an example of these effects in the domain of archaeology see Di Giuseppantonio Di Franco et al. [80]). Depth perception in VR have been demonstrated to reduce time, errors and to improve user's performances involving spatial tasks [277, 365] as well inducing a better performance for object manipulation [198, 364]. However, systematic underestimation of distances was found both with respect to real workspace measurements and to egocentric distances [351, 371, 374].

Tactile augmentation is considered an effective alternative mixed-reality technique for introducing tactile cues [125, 89, 147, 205]. This technique is very effective with dedicated hardware appliances in dedicated exhibit spaces such as CAVE environments, dark rooms, virtual theaters, etc. (for instance, Kenderdine et al. [157]; Forte [90]; Carrozzino et al. [55]). Economic resources and multidisciplinary collaborations are however not always available in order to create and maintain such complex dedicated hardware. To respond to the increased interests of this kind of techniques from museum experts, some scholars have designed systems that allow museum specialists to build and manage virtual and augmented reality exhibitions in an efficient and timely manner, just using a database of 3D models of artifacts [376].

Tactile perception of ancient artifacts is now achievable thanks to advance in technologies, which made it possible to physically reproduce ancient artifacts using 3D printers. Three-dimensional digital copies of artifacts can be printed using Rapid Prototyping (RP) techniques. RP is the process whereby physical objects are created from computer-generated programs (i.e., CAD, 3D Studio Max, etc.) using prototype 3D machines that can build a 3D object out of liquid, solid, or powder material. RP is applied to many fields, such as, architecture, education, healthcare, etc. [67, 42]. Recently, this technique has been used in projects concerned with preservation and reproduction of cultural heritage. For instance, a few companies are now experi-

menting with art museums to 3D print famous paintings with high quality colors, to capture the "physical presence of these paintings" [7, 283]. With the notion of "physical presence", some scholars suggest that texture/relief is as important as colors to understand the uniqueness of a painting. Van Gogh, for instance, used thick layers of colors (i.e., a thick impasto) to create games of lights and shadows in his paintings.

While several works have explored the use of virtual reality replicas or 3D prints in different ways, no study has been performed to date with the specific goal of understanding the advantages and trade-offs in using these modalities for the perception of artifacts. Given the significant recent increase in the number of projects reported in the literature that incorporate 3D digital replicas and/or 3D prints of artifacts, investigating the value of these new technologies for the perception of our past becomes extremely relevant and important. The main contribution of this work is therefore to provide a first study focused on understanding the benefits given by these new technologies. We are not aware of previous work investigating the same questions as the ones addressed in this chapter. The next sessions present our experiments and results.

A.2 Experiment 1

A.2.1 Description of the experiment

In this first experiment we have investigated how people perceive archaeological objects under different interaction modes: (1) visual examination, (2) three-dimensional immersive visualization, and (3) three-dimensional printed replica interaction. This experiment was designed to uncover which medium best enables the perception of the innate qualities of an artifact. In this experiment we have collected information about how people describe and interact with objects reproduced using different media:

1. *Look* (i.e., real-life visual examination) condition: participants viewed objects in a display case of $25x25cm$ located on a table. (Fig. A.1). A caption with information on provenience, age, and size of each object was placed outside the display, 3 cm behind it. The participants in this condition were asked to stand in front of the display windows, look at each object, read the caption, and then, looking at the camera, describe the object and eventually guess the function of this object in the past. The camera was located on the opposite site of the

table (i.e. opposite in relation to the subject). While they were describing the objects participants were left alone in the room.

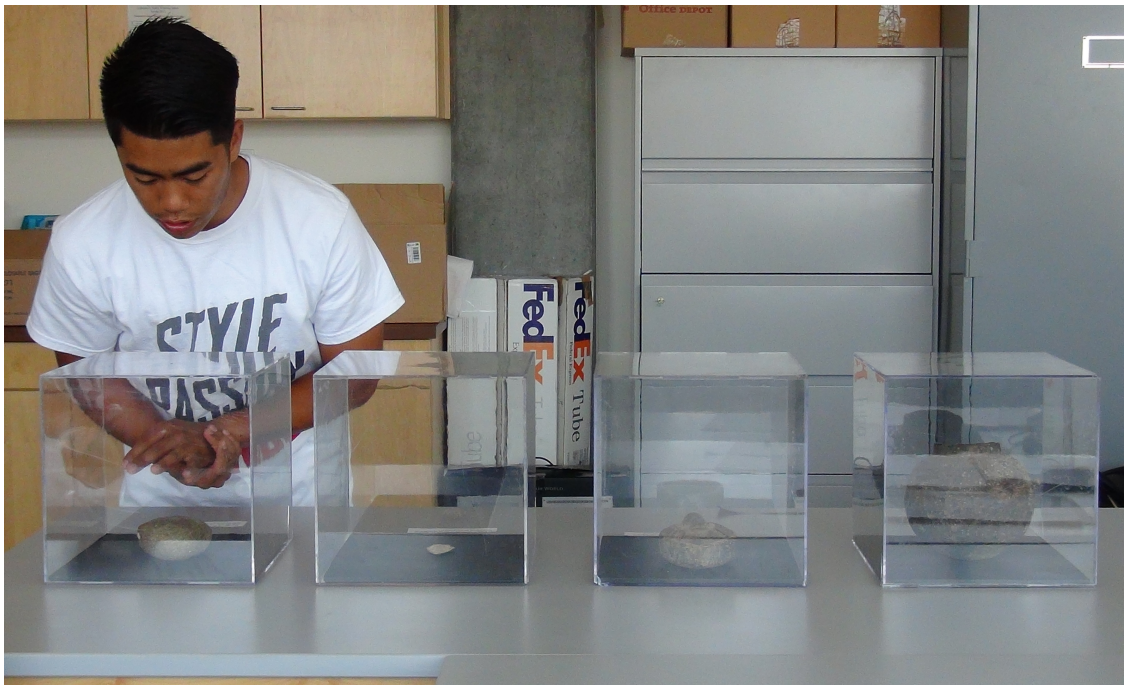


Figure A.1: Participant in the Look condition.

2. *Powerwall* (i.e., 3D immersive visualization) condition: participants interacted with 3D digital copies of objects visualized in an immersive stereovision system (Fig. A.2). The Powerwall is a retro-projected surface of $4.56x2.25m$ illuminated by twelve projectors (each 1024x768@60Hz) with circular passive polarization filters. The projectors are connected to a rendering cluster of six commodity Linux-based rendering nodes (Pentium Q9550 2.83GHz GeForce GTX 280 4Gb RAM) driven by a similar main machine controlling the virtual scene being displayed. The dimensions of the objects and scenes are preserved and perceived by the user as in a real-life experience.

   The 3D digital copies were made using a Next Engine desktop triangulation laser scanner and then optimized and imported in the Powerwall framework (1.6 million triangles and 400 Mb compressed textures in total).

   In this condition, participants were asked to interact with one object at a time and then, when they felt ready, to look at a camera and describe each object

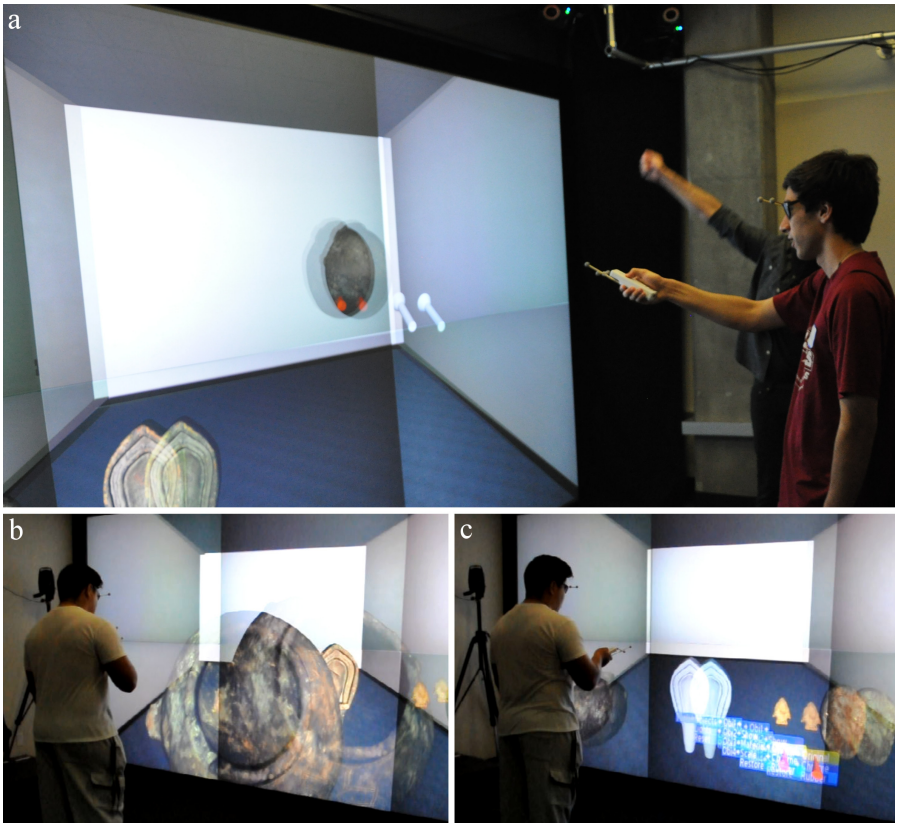


Figure A.2: Powerwall condition. a) Changing light condition to explore objects. b) Manipulating objects (objects appear big on the screen due to off-axis parallax projection but the user perceives it as in real-life); c) Interacting with the objects without original colors (note the floating virtual menu in front of the user).

and then guess its function in the past. Object captions were placed in on a desk close to the participant, in the same appearance order of the objects in the application. The camera was located on the right side of the Powerwall screen, about $2.0m$ from the presenters. In this condition, participants had the option to manipulate the objects interactively and select specific actions through a virtual floating menu. As shown in Figure A.2 the user controls a virtual pointer in the scene (red cone) directly mapped to the position in space of the remote controller. The pointer is perceived by the user as floating in front of the controller being held. The user is able to manipulate each object by selecting it with the virtual pointer, similar to real-life manipulations (Fig. A.2a-b). Through a virtual menu that can be opened and removed at will

(Fig.A.2c), two types of action are possible (Fig. A.3): removing original colors (i.e., texture) to appreciate the 3D model geometry *mesh*, and changing light conditions (environmental or torch light simulation and light source colors). A virtual scale did not accompany the objects displayed during the experiment.

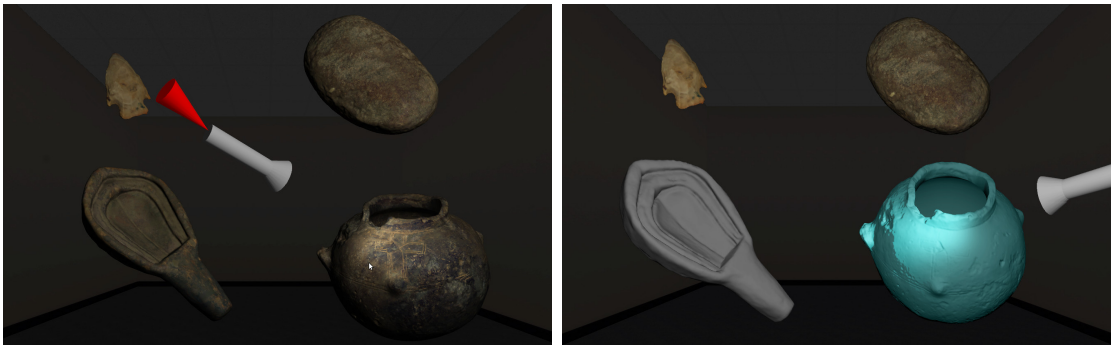


Figure A.3: Highlight of object manipulation and visualization in the Powerwall in dark environmental light condition. The red cone represents the user's pointer designed to interact with the scene (objects and menu 3D interaction and lights repositioning). Left: The user is moving the light source to enhance objects details. Right: similar situation where the objects textures where removed to analyze the polygonal representation.

3. *3D prints* (i.e., 3D printed haptic) condition: participants touched 3D printed copies of original artifacts (Fig. A.4). The prints were located on a table and the caption was placed 3cm behind them. Participants in this condition were asked to take one object at a time in their hands and, while touching the object, describe it looking at the camera, which was on the other side of the table. While they were describing the objects, participants were left alone in the room. The 3D prints were made using a ZCorp rapid prototyping technique, which allows for photo-realistic, color design prints and have a resolution of up to 650x540 DPI. The material used is a powder that is combined with an adhesive and simultaneously embedded with an ink jet print head. Finally, the part can be finished using infiltrants including wax, cyanoacrylate (super glue), and epoxy materials, which increase the 3D parts' strength and create the desired finish to ensure durability and more vivid colors. The printed product is a hard, rigid material that is slightly delicate and not suited for structural parts under great load. While these prints can reproduce size, shape details and color grain with a high level of accuracy, it has some known issues in the reproduction of tonality (the colors are usually faded) and is unable to

reproduce the weight of original objects. Nonetheless, by fortune, the only 3D print, whose weight consistently differed from the weight of the original (about three times heavier), is the Buddhist one. In this case, the original artifact is made of a considerably light type of wood.

Figure A.4: Participant in the 3D prints condition.

Sixty people participated in this study. All were undergraduate students who received extra credit in a class. Half the participants were female. All were highly proficient English speakers with normal or corrected vision.

Participants either in the Look or 3D prints conditions were left alone in a lab facility, free to interact with artifacts displayed using a unique medium and then completed a questionnaire to explain their experience with each object. Participants in the Power Wall condition were left alone in the Virtual Reality lab, in front of the Powerwall. After they interacted with the 3D digital replicas they completed a questionnaire to explain their experience with each object. The questionnaires were analyzed in order to determine which type of interaction (physical or virtual) best serves research and presentation needs of archaeological material to the general

public. Each participant was in only one condition, for instance, only in the Look condition or only in the 3D print condition.

Four artifacts made from a range of different materials and coming from different geographic areas and chronological contexts, were selected for the experiment, with the aim of evaluating the degree to which the techniques of 3D scanning and printing are perceived differently for different materials (e.g., stone, pottery, etc.), shape, and other physical qualities such as weight, density, and so on. The artifacts selected were: a. Buddhist ritual object from Nepal; b. grinding stone from California; c. ceramic vessel from Ethiopia; d. projectile point from California (Figs. A.5A.6A.7).

Figure A.5: Objects selected for the experiment: a. Buddhist ritual object from Nepal; b. Grinding stone from California; c. Ceramic vessel from Ethiopia; d. Projectile point from Californian.

Figure A.6: 3D prints of the objects selected for the experiment: a. Buddhist ritual object from Nepal; b. Grinding stone from California; c. Ceramic vessel from Ethiopia; d. Projectile point from California.

Below we report a few of the most interesting findings we observed in our data.

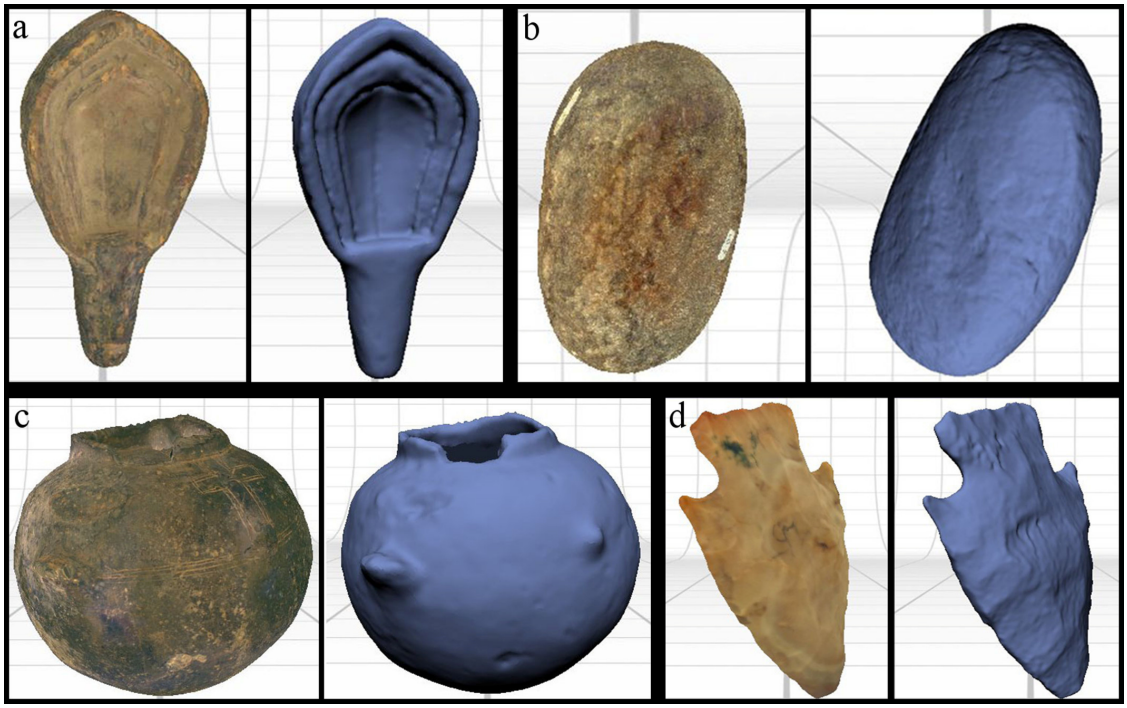


Figure A.7: 3D virtual reproductions of the objects selected for the experiment: a. Buddhist ritual object from Nepal; b. Grinding stone from California; c. Ceramic vessel from Ethiopia; d. Projectile point from California.

A.2.2 Results

We conducted an analysis of responses using one-way ANOVA with the three between-subjects perceptual condition factors (individual comparisons where performed through Tukey's HSD and Bonferroni tests). The ANOVA analysis compares mean differences among three or more experimental conditions. In this experiment the null hypothesis states that the means of all conditions are not statistically different from one another. The null hypothesis is rejected when at least one of the means being compared is significantly different from the others, which is indicated by a resulting p-value of less than .05. We used one-way ANOVA for each of the following questions (see table A.1 for mean values and standard deviations):

Q1. *How heavy is this object compared to an apple? (likert scale with 1 being "very heavy" and 9 being "very light")*

Overall, for the case of the grinding stone, $F(2, 57) = 4.38; p = .017$, participants

in the 3D prints condition perceived the objects heavier than participants in the Look condition. In addition, looking at the trend proposed by the other objects a similar pattern can be recognized. Participants in the Powerwall condition perceived the objects' weight similarly, but not significantly, than participants in the Look Condition.

Q2. *How easy was to appreciate the colors of this object? (Likert scale with 1 being very difficult, and 9 very easy)*

Participants in the Look and Powerwall condition found it easier to perceive the colors of the objects than participants in the 3D prints condition. The difference was found significant only considering the results from the data retrieved from the Projectile Point, $F(2, 57) = 3.61; p = .034$. However, even in this case, all the means where similarly showing the same pattern.

Q3. *How big is this object compared to an apple? (Likert scale with 1 being very small and 9 being very large)*

Participants in the Powerwall condition perceived both the Buddhist ritual object, $F(2, 57) = 4.79; p = .012$, and the Grinding stone, $F(2, 57) = 3.91; p = .026$, smaller than participants in the Look condition. A similar trend can be seen considering the case of the the 3D prints condition where participants also perceived both the Buddhist ritual object and the Grinding stone bigger than participants in the Powerwall condition, but in this case the difference is not significant. Even for the Ceramic vessel a similar tendency can be seen even though, as shown by the projectile point data, participants in all conditions selected similar values (Average: Proj. point 1.2-1.7; Ceramic vessel 8.1-8.7) to define the size of these objects.

Q4. *What is the texture of this object? (Likert scale with 1 being smooth and 9 rough)*

Participants in the Look condition perceived the Projectile point as significantly smoother than participants in the Powerwall and 3D prints conditions, $F(2, 57) = 4.41; p = .017$. This result seems to be in contrast with the tendency shown by the other two objects, nonetheless the tendency was not statistically significant. Conclusions are summarized in Section A.6.

Table A.1: Mean values and standard deviations of the Likert scale questionnaire (a. Buddhist Ritual object; b. grinding stone; c. ceramic vessel; d. projectile point).

Conditions							
		Powerwall		*3D Prints*		*Look*	
Questions		**M**	**SD**	**M**	**SD**	**M**	**SD**
Q1	*a*	6.6	1.63	6.8	1.61	6.65	1.63
	b	7.45	2.01	6.65	1.35	8.05	.94
	c	7.3	1.56	6.8	1.76	7.4	1.76
	d	2.0	1.17	1.35	0.49	1.6	1.09
Q2	*a*	7.1	2.02	6.75	2.17	7.15	2.03
	b	7.4	2.04	6.15	2.54	7.05	2.63
	c	7.0	2.34	6.65	2.41	7.1	1.97
	d	6.95	1.67	6.75	2.09	8.15	1.53
Q3	*a*	6.65	1.09	7.4	1.14	7.7	1.08
	b	5.9	1.68	6.65	1.14	7.05	1.05
	c	8.15	0.99	8.45	0.76	8.7	0.80
	d	1.65	1.04	1.4	0.59	1.25	0.55
Q4	*a*	5.7	2.11	5.2	1.61	6.35	1.76
	b	5.55	2.66	5.3	2.05	6.05	1.7
	c	5.9	2.75	5.0	2.38	6.3	2.41
	d	5.95	2.42	5.75	2.19	3.9	2.59

A.3 Experiment 2

A.3.1 Description of the experiment

In the second experiment we have examined how people use gestures to describe objects in different modes: (1) traditional visual examination, (2) 3D immersive visualization, and (3) 3D printed replica interaction. The goal of this second experiment was to analyze when and how gestures were used in discourse about artifacts displayed in varied media.

We had participants interacting with objects in the same conditions as described in the previous experiment: Look, Powerwall, and 3D prints. Thirty people participated in the study (the number of participants was determined based on previous studies; e.g., [215]). All were undergraduate students who received extra credit in a class. Half the participants were female. All were highly proficient English speakers with normal or corrected vision.

Participants were video recorded during the experiments (in the Virtual Reality lab or in another lab) and before starting each activity they completed two surveys: a demographic survey (age, major area of study, etc.) and a survey about their previous experience with artifacts (real or digital). After the surveys were completed

they were given verbal instructions and then were left alone during the experiment, in order to let them feel more comfortable in front of the camera.

Interviews were video recorded (with audio). The gestures in the videos were analyzed in order to determine which type of interaction condition is most suitable for research. Our analysis compared how participants gestured while talking about the artifacts. Gestures are believed to facilitate reasoning and learning [102, 215] and can help in describing abstract objects [28]. Gesture scholars often distinguish between beat gestures and iconic gestures. Beat gestures are rhythmic hand movements that convey no semantic information, but are believed to facilitate lexical access [173].

When describing an artifact, for instance, a person might make three short repeated gestures to help formulate what she is trying to say (e.g., shaking one hand). Iconic gestures are manual movements that convey visual-spatial information about the topic of discourse [220]. While describing the function of a grinding stone, for instance, a person might say, "this is for grinding corn", while making a gesture that depicts the action of grinding.

Each subject participated in only one condition. Below we report a few of the most interesting findings we have observed in our data.

A.3.2 Results

Our in-depth analysis examined when and how iconic and beat gestures were used in discourse about artifacts displayed in varied media. Table A.2 shows the values for average number of gestures produced by each group of participants in each condition. Participants produced more iconic gestures in the 3D prints condition and fewer in the Powerwall condition, but the difference was not significant.

Participants used more beat gestures in the Powerwall condition than in all other conditions. This finding was reliable when comparing Powerwall to both Look and 3D prints conditions, $F(2, 27) = 4.31; p = .024$.

Table A.2: Mean and standard deviation of beat and iconic gestures produced by participants while talking about the artifacts.

	Beat		**Iconic**	
Conditions	**M**	**SD**	**M**	**SD**
Powerwall	28.1	13.75	3.9	3.48
3D prints	8.1	18.42	5.9	3.51
Look	7.8	6.23	5.3	4.69

Subsequently, we classified types of iconic gestures used by participants while describing the artifacts. Gestures were mainly used to describe motion. Iconic gestures conveying motion were frequently used to give information about the function of an object. For instance, while talking about a projectile point, a few participants said: "It was used for hunting" and then mimicked the action of throwing a spear or dart to kill an animal. Similarly, while describing a grinding stone, some participants mimicked the circular motion performed by people to grind seeds or other vegetal foods. Gestures included describing the original context in which the object was likely used; for instance, some people visually described the shape of a metate (i.e., milling slab) in association with the grinding stone (believed to be a mono) or associated the latter to the Buddhist object, when this was believed to be a metate (Fig. A.8a).

Participants often used gestures while talking about how the artifact was manufactured; for example, while describing the projectile point, a few participants simulated the flaking process. Iconic gestures were also used to define the shape of an object and/or stress elements of shape (Fig. A.8b). In the case of a pot, an object missing part of the lip and handle, gestures helped to stress the shape of the missing parts. Some participants performed iconic gestures while talking about textures and materials of an object. Iconic gestures also helped some people convey the size of an object, especially in cases where it was difficult to determine object scale (Fig. A.8c).

A few other observations on how participants interacted with various media are in order. All participants in the Look condition seemed more uncomfortable when interacting with artifacts than their peers in the other conditions. In viewing the objects displayed in cases, they often leaned close to examine specific details. At the same time, though, they kept their hands far from the case. Some participants put their hands behind their back, and others, rested their hands on the table. Some participants shyly touched cases with their fingertips and then quickly retracted them.

Participants in the Powerwall condition could interact with 3D replicas of artifacts with the remote controller. They were able to virtually manipulate the artefact before describing it, but they were asked not to touch the remote controller while talking. Observing the videos, we noticed that during the stage of interaction with the artifacts (i.e., before talking) most participants behaved as if they were touching the objects (i.e., as if the objects were "real", holding the object with the remote controller while touching it with the free hand). However, even though instructed, while talking about these objects in front of the camera, participants found it difficult

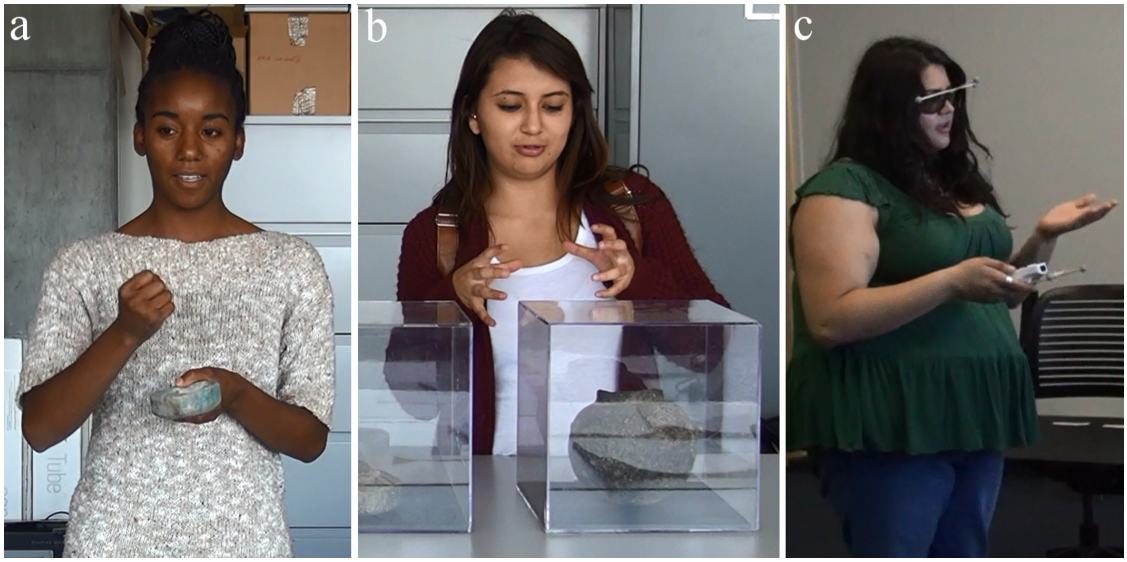


Figure A.8: Iconic gestures performed while describing the artifacts. A. describing the function of the grinding stone (mono) in association with the Buddhist object (considered to be a metate); b. describing the shape of the ceramic vessel; c. defining the size of the Buddhist object (compared to a hand).

not to touch the remote controller. Finally, 3D print participants interacted with 3D prints as they would with real-life objects.

For interpretations of these results see discussions and conclusions (Sections A.5 and A.6).

A.4 Experiment 3

A.4.1 Description of the experiment

In April of 2014, we organized an exhibition titled "What are you 'Looking' at: Experiencing Ancient Artifacts". Through hands-on 3D virtual and material interaction with ancient artifacts, the exhibition was aimed at problematizing the archaeological display and showing how our perception of the past is affected by the medium used to present it.

All participants were first brought to the Powerwall room (stage 1), where they interacted with 3D digital replicas of artifacts through the immersive system (see

Figure A.9: Participants trying to touch 3D objects the objects on the Powerwall (exhibition, stage 1).

experiment 1; Fig. A.9). In a second stage (stage 2), all participants were guided to another room where they saw the original artifacts displayed in glass cases and also interacted with pictures, 3D prints, and 3D digital replicas of the same artifacts displayed on a computer screen. In this room, they were free to interact with one of the medium and were then asked to voluntarily participate in a questionnaire and rate (Likert scale) their overall experience with both the Powerwall and the other medium chosen (Fig. A.10).

Sixty visitors agreed to participate in the questionnaire. During stage 2, just a few participants selected the 3D digital replicas on the PC (4 out of 60), while no one wanted to interact with the pictures. For this reason the 3D digital replicas and pictures were not included in the statistical analysis related to evaluate participants' engagement with the medium (see the following section).

A.4.2 Results

The rating scores were transformed in mean scores (see Table A.3) and correlated using ANOVA statistical analysis.

We first compared all questions in order to analyze to which extent the medium helped visitors to understand the characteristics of the artifacts. Comparisons between Q1 (lights settings in the Powerwall), Q6 (Tactile experience with 3D prints),

Figure A.10: Participants interacting with original artifacts inside cases (left) and with 3D prints (right; exhibition stage 2).

and Q10 (visual experience with original objects) revealed no statistical difference between the three conditions. However, looking at the means, we noticed that while these values almost coincided when observing Powerwall and Look conditions, they are slightly higher in the 3D prints condition (i.e., the tactile experience was rated higher).

When comparing Q2 (removing color from digital artifacts in the PW) to Q6 (Tactile experience with 3D prints), and Q10 (visual experience with original objects) revealed statistical difference between 3D prints and Powerwall conditions, $F(2, 54) = 3.52; p = .037$.

In summary, the possibility of changing light settings in the Powerwall was considered almost as useful as touching 3D prints or looking at original artifacts for the understanding of the objects' physical qualities. On the other hand, the ability of removing original colors from the 3D digital models was not considered as effective as touching 3D prints.

Second, we compared all questions in order to analyze which of the three conditions/media participants considered most helpful for the artifacts' understanding (Q3, Q7, Q11). Statistical analysis showed no reliable difference between the three conditions. However the mean values for Powerwall and 3D prints conditions were higher suggesting how these conditions were considered slightly more helpful than Look to appreciate the artifacts.

Table A.3: Questionnaire Experiment 3 excerpt. Likert scale with 1 being Strongly Disagree and 9 being Strongly Agree.

Powerwall

	M	**SD**
Q1. The possibility to select appropriate lights improved my understanding of the artifacts characteristics	7.45	1.54
Q2. The possibility to remove original colors of the artifacts improved my understanding of the artifacts characteristics	6.5	2.37
Q3. The ability to use the Powerwall (full scale 3D screen) was very helpful compared to a traditional museum display	7.8	1.7
Q4. The Powerwall system seems to be a good approach to interact with ancient artifacts	8.2	1.11
Q5. This experience with 3D digital artifacts was engaging	8.9	0.31

3D Prints

Q6. The possibility to touch 3D printed artifacts improved my understanding of the artifacts characteristics	8.00	1.08
Q7. The ability to interact with 3D printed artifacts was very helpful compared to interacting with 3D digital artifacts in the Powerwall	7.6	1.5
Q8. 3D prints seem to be a good approach to interact with ancient artifacts	8.05	1.05
Q9. This experience with 3D prints was engaging	8.3	0.86

Look

Q10. The possibility to look at original artifacts through a display improved my understanding of the artifacts' characteristics	7.47	1.74
Q11. The ability to look at the artifacts was very helpful compared to interacting with 3D digital copies in the Powerwall	6.64	1.98
Q12. Traditional display seems to be a good approach to interact with ancient artifacts	7.35	2.12
Q13. This experience with original artifacts was engaging	7.12	2.20

Third, comparing questions aiming at rating the overall effectiveness of each medium (Q4, Q8, Q12), we did not find any reliable difference. However, on average the Powerwall and 3D prints were considered slightly more effective than Look to interact with ancient artifacts.

Finally, when comparing all questions aimed at rating engagement within each condition, we found that the Powerwall and 3D prints conditions were considered significantly more engaging than the Look condition, $F(2, 54) = 8.58; p = .001$. The questionnaire ended with a multiple choice question in which we asked participants to compare the experience they had with the Powerwall to that they had with the medium selected during stage 2, and an open-ended question in which we asked to explain why they preferred a particular experience. As mentioned before pictures and 3D replicas on PC screen were not included in the analysis, since just a few participants interacted with these two media (3D digital copies on a PC: 4 out of 60; pictures: 0). Yet, it is interesting to notice that 3 out of the 4 participants who interacted with the 3D digital copies on the PC screen preferred the Powerwall

experience and one was neutral.

Comparisons between Powerwall and the remaining conditions (Look and 3D prints) revealed that participants interacting with original artifacts exhibited in glass cases preferred the experience with the Powerwall, $X2(2, N = 18) = 2.12, p = 0.03$. Most of the participants who expressed their preference for 3D prints and Powerwall explained that these experiences were more engaging because they could touch (i.e., with the 3D prints) or "almost" touch the objects (i.e. in the Powerwall).

A.5 Discussion

The presented studies investigated how different presentation modalities influence the understanding of artifacts. We were especially interested in how people would interact with, understand, and describe objects with three different conditions: visual experience with authentic artifacts, 3D digital reconstructions in the Powerwall, and 3D prints. Participants described ancient artifacts in front of a video camera.

The results from our experiments show how the different presentation modalities affect the perception of different characteristics of the objects. With respect to weight information, our findings show that, in an immersive 3D reality situation participants perceive objects' weight similarly to what people would perceive in a museum (i.e., looking at original artifacts located in a case). In both cases the weight estimation relies on pure visual cues that, in our opinion, would force the participant to think about the original material more carefully. Moreover, similarly to the discussion presented by [119], Immersive VR systems expose users to visual cues that make it difficult to estimating the weight of an object. In the VR medium the weight estimation is similar to the real looking scenario. Using 3D Prints the participant may have based its judgment on the actual weight of the object held. However, because of the unavailability of the original artifacts we could not compare the weight estimation of the three media with an estimation of the weights from the originals.

With regard to color information (color grain, variation, and tonality) of the artifacts selected for the experiment, Powerwall and Look conditions give a similar level of perception, indicating the ability of the Powerwall system to well display this kind of information. This finding is reinforced by the fact that participants in experiment 2 indicated light variation as an effective means to perceive and understand the artifacts.

With respect to size, the Look, Powerwall and 3D prints conditions show very

similar results for both the ceramic vessel and the projectile point, which have a size not at all close to that of the reference object (an apple). For the grinding stone and the Buddhist objects instead, whose size is closer to that of the reference point, statistical analysis shows how these two objects are considered significantly smaller in the Powerwall than in the Look condition. This finding reinforces the idea that distance and size misestimation in immersive virtual environments is higher than in real scenarios (Naceri et al. [236] & Thompson et al. [351]), even for virtual reconstruction of archaeological objects.

Regarding texture qualities, the projectile point is the only one of the objects used for the experiment for which we found a reliable difference when we compared participants in the Powerwall and 3D prints conditions to participants in the Look condition. The latter participants, in fact, perceived this object as considerably smoother than their peers in the other two conditions. Our finding suggests that in presence of small, bright, and light-colored objects, visual cues are not enough to accurately perceive texture qualities. Based on this finding while participants in the Look condition could grasp the sense of texture of the objects only based on visual cues, participants in the Powerwall could rely on multi-visualization tools, such as different light settings and the possibility to zoom in and remove original colors from the 3D models, to grasp textural information. To reinforce this statement we found that more than one participant stressed the importance of removing colors and changing light settings for perceiving texture qualities. One participant said: "watching the chrome object [i.e., object without original colors], I was able to see different, other details that I was not able to see with the original colors" (emphasis on the original interview).

The qualitative analysis of gestures (experiment n. 2) shows that, in absence of a tactile experience, people produce some stereotypical iconic gestures to mimic actions they would perform if they were actually touching the artifacts. The iconic gestures performed often convey spatial information; they help people mimic object manufacturing and function. Gestures can also be used to describe details of shape and also help people figure out the size of an object.

As noted, when people described objects they also produced beat gestures (which do not convey any meaning per se). The results of this experiment show how participants looking at original artifacts inside cases generated the fewest gestures.

Conversely, participants interacting with objects in the Powerwall used the highest number of beat gestures. The high number of beat gestures was reliably different from the number of gestures produced by participants in the Look and 3D prints condition. The difference with the 3D prints does not really surprise, since partic-

ipants were talking while holding the objects, thus it was more difficult for them to perform gestures. What is more surprising is the difference between Look and Powerwall conditions. In both cases participants had their hands free while talking. It is possible that the cases represented a psychological barrier that inhibited participants' direct experience with the objects. This idea is reinforced by the fact that, when they interacted with these objects, they kept their hands far from the case (i.e., they seemed afraid of touching it) (see Figure A.2). Conversely, following Krauss [173], who argued that beat gestures often facilitate lexical access, it is possible that the high number of beat gestures reflects a lack of certainty about artifact details (i.e., participants were less certain about what they were talking about, but it might also indicate that in the immersive system participants recognized a difference, a frame, between the physical and the virtual world and tried to fill this gap using gestures. Another suggestive explanation, which would need further analysis, might be linked with the engagement participants had while interacting with the Powerwall, how demonstrated by the results of Experiment 3. In that case the high number of beat gestures might be directly correlated with the excitement people had while interacting with the Powerwall.

Experiment 3 was mainly designed to collect metacognitive information on how useful the participants considered each experience for the perception and understanding of the artifacts proposed and how engaging the experience with each condition was. Overall Powerwall and 3D prints were considered more helpful and more engaging than the visual experience with real artifacts.

A.6 Conclusions

We presented three experiments aimed at understanding how people, interact, perceive and engage with past material culture in different media states. Our results demonstrate the potential of new technologies and help design best practices and design choices for improving museum displays in museums and other exhibitions.

Results from experiment 1, which focused on the perception of specific characteristics of ancient artifacts in different media states, revealed that the media selected for the experiment affect the perception of physical qualities of artifacts in different ways. The immersive experience with the Powerwall and visual experience with original artifacts resulted in similar perception patterns for color and weight; these characteristics are difficult to perceive with the 3D prints; as a result, the misinterpretation of weight and color might also bring to misinterpretation of other qualities (e.g., material) and of the function of the artifacts. While experiencing the objects

in the powerwall resulted in size misinterpretation, it was a useful mean to recognize texture qualities, especially for small and bright objects.

Results from experiment 2, aimed at investigating how we describe and interact with ancient artifacts through our body, suggests that traditional museum settings may diminish or limit museum users degree of engagement with ancient artifacts.

The latter finding seems reinforced from the results of experiment 3, which give us insights into people's engagement with artifacts through different media. These results suggest that, in absence of a tactile experience with original artifacts, our sample of participants favored a tactile or semi-tactile experience with replicas to the visual experience with original ancient objects. In other words, these participants were ready to negotiate with the inauthentic in order to have a tactile embodied experience. Even though some of these results might seem obvious to scholars who design and test immersive systems, they can be noteworthy for scholars in the heritage, archaeology and museum domains. This is because ancient artifacts are a unique type of objects, which carry information about past cultures. Thus, we expected that authentic artifacts displayed in a case would trigger emotions that 3D copies (virtual and real) could not equal. On the contrary our findings show that the conditions Powerwall and 3D prints were most appreciated, suggesting how our sample of participants are more concerned with experiencing an object through the senses rather than having the original in front of them. Similar findings have been reported by other studies ([380, 222, 352]).

Our findings bring us to re-consider how we approach museum displays today, since visitors seem to claim for an active experience with the past, which emphasizes a kinesthetic engagement with the museum environment. These findings also suggest that although new technologies are not yet able to reproduce in full the perception that people would have manipulating original artifacts, these technologies produce excitement and engagement, encouraging curiosity, attention, and desire for knowledge of past material culture.

Our study represents a starting point for the creation of a protocol or methodology that envisages the integration of different technologies within a museum. It would be interesting, for instance, to see what happens to perception, engagement, and understanding if visitors interact with an object in a 3D immersive environment, or through a 3D print first, and then visit the showcase in which the original counterpart is showcased.

In sum this analysis shows that people like to engage with new technologies to understand ancient artifacts and point to the integrated use of traditional display, 3D immersive systems, and 3D prints as an effective way to increase perception, un-

derstanding, and engagement with artifacts, as well as favoring a diverse population of museum visitors.

While our current work uncovers some first observations in this area, there is plenty of further development worth exploring. It would be critical, for instance, to investigate what may be influences (ethnicity, gender, education, socio-economic background) in varying perceptions of authenticity in relation to objects, virtual and real. And importantly, how these results might vary across cultures. It would also be interesting to explore how people with particular affiliation with tangible heritage interact with both authentic objects and their reproductions in different media states.

References

[1] 5dt. *5dt Data Gloves*, 2014 (accessed September 20, 2014). *http://www.5dt.com*.

[2] S.J. Ahn and J.N. Bailenson. Self-endorsing versus other-endorsing in virtual environments. *Journal of Advertising*, 40(2):93–106, 2011.

[3] S.J. Ahn, J.N. Bailenson, and D. Park. Short-and long-term effects of embodied experiences in immersive virtual environments on environmental locus of control and behavior. *Computers in Human Behavior*, 39:235–245, 2014.

[4] J. Ahrens and J. Painter. Efficient sort-last rendering using compression-based image compositing. In *in Proceedings of the 2nd Eurographics Workshop on Parallel Graphics and Visualization*, pages 145–151, 1998.

[5] G. Alankus, R. Proffitt, C. Kelleher, and J. Engsberg. Stroke therapy through motion-based games: A case study. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pages 219–226, New York, NY, USA, 2010. ACM.

[6] A. Alaraj, M. Lemole, J. Finkle, R. Yudkowsky, A. Wallace, C. Luciano, P. Banerjee, S. Rizzi, and F. Charbel. Virtual reality training in neurosurgery: Review of current status and future applications. *Surgical Neurology International*, 2(1):52, 2011.

[7] D. Alberge. Van Gogh in 3D? A replica could be yours for 22,000. Museum develops hi-tech replicas of Dutch master accurate right down to the frame, 2014 (accessed September 20, 2014). *http://www.theguardian.com/artanddesign/2013/aug/24/3d-replicas-van-gogh*.

[8] P.L. Alfano and G.F. Michel. Restricting the field of view: perceptual and performance effects. *Perceptual and motor skills*, 70(1):35–45, Feb 1990.

[9] J. Allard, V. Gouranton, L. Lecointre, E. Melin, and B. Raffin. Net juggler: running vr juggler with multiple displays on a commodity component cluster. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 273–274, 2002.

[10] J. Allard, C. Ménier, E. Boyer, and B. Raffin. Running large vr applications on a pc cluster: The flow vr experience. In *Proceedings of the 11th Eurographics*

Conference on Virtual Environments, EGVE'05, pages 59–68, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[11] F. Anderson, T. Grossman, J. Matejka, and G.W. Fitzmaurice. YouMove: enhancing movement training with an augmented reality mirror. In *Proceedings of User Interface Software and Technology (UIST)*, pages 311–320. ACM, 2013.

[12] B.D. Argall, B. Browning, and M. Veloso. Learning by demonstration with critique from a human teacher. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 57–64. ACM, 2007.

[13] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009.

[14] O. Arikan and D.A. Forsyth. Synthesizing constrained motions from examples. *ACM Transactions on Graphics*, 21(3):483–490, 2002.

[15] O. Arikan, D.A. Forsyth, and J.F. O'Brien. Motion synthesis from annotations. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 402–408, New York, NY, USA, 2003. ACM.

[16] C. Armbruster, M. Wolter, T. Kuhlen, W. Spijkers, and B. Fimm. Depth perception in virtual reality: distance estimations in peri and extrapersonal space. *Cyberpsychology Behavior*, 11(1):9–15, Feb 2008.

[17] S. Arteaga, J. Chevalier, A. Coile, A.W. Hill, S. Sali, S. Sudhakhrisnan, and S.H. Kurniawan. Low-cost accelerometry-based posture monitoring system for stroke survivors. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 243–244, New York, NY, USA, 2008. ACM.

[18] K.W. Arthur. Effects of field of view on performance with head-mounted displays. Doctor of Philosophy, Computer Science. The University of North Carolina at Chapel Hill, 2000.

[19] American Physical Therapy Association. *Guide to Physical Therapist Practice. Rev 2nd Ed.* American Physical Therapy Association, Alexandria, VA, 1999.

[20] ATI. ATI Eyefinity Technology, 2014 (accessed September 20, 2014). *http://www.amd.com/en-us/innovations/software-technologies/eyefinity*.

[21] A.P. Atkinson, W.H. Dittrich, A.J. Gemmell, and A.W. Young. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception*, 33(6):717–746, 2004.

[22] N.I. Badler, C.B. Phillips, and B.L. Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, Inc., New York, NY, USA, 1993.

[23] M. Balaam, S. Rennick Egglestone, G. Fitzpatrick, T. Rodden, A. Hughes, A. Wilkinson, T. Nind, L. Axelrod, E. Harris, I. Ricketts, et al. Motivating mobility: designing for lived motivation in stroke rehabilitation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3073–3082. ACM, 2011.

[24] B.D. Balaguer and S. Carpin. A learning method to determine how to approach an unknown object to be grasped. *International Journal of Humanoid Robotics*, 8(03):579–606, 2011.

[25] R. Ball and C. North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1196–1199, New York, NY, USA, 2005. ACM.

[26] A.G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[27] D. Bartz, X. Pueyo, and E. Reinhard. Out-of-core sort-first parallel rendering for cluster-based tiled displays. In *in Proceedings of the 4th Eurographics Workshop on Parallel Graphics and Visualization*, 2002.

[28] J. B. Bavelas, N. Chovil, D. A. Lawrie, and A. Wade. Interactive gestures. *Discourse processes*, 15(4):469–489, 1992.

[29] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta. Assessment in and of serious games: An overview. *Advances in Human-Computer Interaction*, 2013:1:1–1:1, January 2013.

[30] S. Benford, J. Bowers, L. E. Fahln, C. Greenhalgh, J. A. Mariani, and T. Rodden. Networked virtual reality and cooperative work. *Presence*, 4(4):364–386, 1995.

[31] H. Benko, E.W. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Mixed and Augmented Reality, 2004.*

ISMAR 2004. Third IEEE and ACM International Symposium on, pages 132–140, Nov 2004.

[32] M. Bergamasco. Le musee des formes pures. In *8th IEEE International Workshop on Robot and Human Interaction, RO-MAN*, volume 99, pages 27–29, 1999.

[33] P. Bhaniramka, P.C.D. Robert, and S. Eilemann. Opengl multipipe sdk: a toolkit for scalable parallel rendering. In *Visualization, 2005. VIS 05. IEEE*, pages 119–126, Oct 2005.

[34] A. Bierbaum and C. Just. Software tools for virtual reality application development. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 3.1–3.45, New York, NY, USA, July 1998. ACM.

[35] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira. Vr juggler: A virtual platform for virtual reality application development. In *Proceedings of the Virtual Reality 2001 Conference (VR'01)*, VR '01, pages 89–, Washington, DC, USA, 2001. IEEE Computer Society.

[36] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1371–1394. Springer Berlin Heidelberg, 2008.

[37] A. Billard and M.J. Matarić. Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37:2-3:145–160, November, 30 2001.

[38] R. Blake and M. Shiffrar. Perception of Human Motion. *Annual Review of Psychology*, 58(1):47–73, 2007.

[39] J. Blascovich and J.N. Bailenson. *Infinite Reality - Avatars, Eternal Life, New Worlds, and the Dawn of the Virtual Revolution.* William Morrow, New York, 2011.

[40] B. Bonnechere, B. Jansen, P. Salvia, H. Bouzahouene, L. Omelina, J. Cornelis, M. Rooze, and S. Van Sint Jan. What are the current limits of the kinect sensor. In *Proc 9th Intl Conf. Disability, Virutal Reality & Associated Technologies, Laval, France*, pages 287–294, 2012.

[41] R. Boulic and D. Raunhardt. Integrated analytic and linearized inverse kinematics for precise full body interactions. In Arjan Egges, Roland Geraerts, and Mark Overmars, editors, *Motion in Games*, volume 5884 of *Lecture Notes in Computer Science*, pages 231–242. Springer Berlin Heidelberg, 2009.

[42] S. Bradshaw, A. Bowyer, and P. Haufe. The intellectual property implications of low-cost 3d printing. *ScriptEd*, 7(1):5–31, 2010.

[43] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A.L. Thomaz. Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems*, 54(5):385–393, 2006.

[44] O. Breeben. *Introduction to Physical Therapy for Physical Therapy Assistants*. Jones and Barlett, Sudbury, MA, 2007.

[45] H. Brenton, M. Gillies, D. Ballin, and D. Chatting. The uncanny valley: does it exist. In *19th British HCI Group Annual Conference: workshop on human-animated character interaction*, 2005.

[46] S.A. Brewster. The impact of haptic touchingtechnology on cultural applications. In *Proceedings of EVA*, volume 1, pages 1–14, 2001.

[47] Jurgen Broeren, Martin Rydmark, and Katharina Stibrant Sunnerhagen. Virtual reality and haptics as a training device for movement rehabilitation after stroke: a single-case study. *Archives of physical medicine and rehabilitation*, 85(8):1247–1250, 2004.

[48] A. Bruderlin and L. Williams. Motion signal processing. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 97–104, New York, NY, USA, 1995. ACM.

[49] J.W. Burke, M. McNeill, D. Charles, P. Morrow, J. Crosbie, and S. McDonough. Serious games for upper limb rehabilitation following stroke. In *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications*, VS-GAMES '09, pages 103–110, Washington, DC, USA, 2009. IEEE Computer Society.

[50] P. Buttolo, P. Stewart, and Y. Chen. Force-enabled sculpting of cad models. *Proceedings of ASME (American Society of Mechanical Engineers) IMECE2000, International Mechanical Engineering Congress and Exposition*, 2000.

[51] S. Calinon. *Robot programming by demonstration*. EPFL Press, 2009.

[52] S. Calinon and A. Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, HRI '07, pages 255–262, New York, NY, USA, 2007. ACM.

[53] M. Cameirao, B. Bermudez, and P. Verschure. Virtual reality based upper extremity rehabilitation following stroke: a review. *Journal of CyberTherapy & Rehabilitation*, 1(1):63–74, 2008.

[54] C. Camporesi, Y. Huang, and M. Kallmann. Interactive motion modeling and parameterization by direct demonstration. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, IVA'10, pages 77–90, Berlin, Heidelberg, 2010. Springer-Verlag.

[55] M. Carrozzino and M. Bergamasco. Beyond virtual museums: Experiencing immersive virtual reality in real museums. *Journal of Cultural Heritage*, 11(4):452–458, 2010.

[56] J. Cassell. Embodied conversational agents: Representation and intelligence in user interfaces. *Artificial Intelligence Magazine*, 22(4):67–83, October 2001.

[57] CAVELib. CAVELib virtual reality software, 2014 (accessed September 20, 2014). *http://www.mechdyne.com/cavelib.aspx*.

[58] X. Cavin, C. Mion, and A Filbois. Cots cluster-based sort-last rendering: performance evaluation and pipelined implementation. In *Visualization, 2005. VIS 05. IEEE*, pages 111–118, Oct 2005.

[59] J. Chai and J.K. Hodgins. Performance animation from low-dimensional control signals. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 686–696, New York, NY, USA, 2005. ACM.

[60] T. Chaminade, J.K. Hodgins, and M. Kawato. Anthropomorphism influences perception of computer-animated characters's actions. In *Social Cognitive and Affective Neuroscience*. Books (MIT Press, 2007.

[61] C. Chang. The design of a shoulder rehabilitation game system. In *2010 IET International Conference on Frontier Computing. Theory, Technologies and Applications*, pages 151–156, 2010.

[62] Y. Chang, S. Chen, and J. Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32(6):2566 – 2570, 2011.

[63] A. Chella, H. Dindo, and I. Infantino. A cognitive framework for imitation learning. *Robotics and Autonomous Systems*, 54(5):403–408, 2006.

[64] J. Chen, H. Cai, A. P. Auchus, and D. H. Laidlaw. Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2130–2139, 2012.

[65] J. Chen and A. Zelinsky. Programing by demonstration: Coping with suboptimal teaching actions. *The International Journal of Robotics Research*, 22(5):299–319, 2003.

[66] C.C. Chiu and S. Marsella. A style controller for generating virtual human behaviors. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '11, pages 1023–1030, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[67] C.K. Chua, K.F. Leong, and C.S. Lim. *Rapid Prototyping: Principles and Applications*. World Scientific, 2010.

[68] A. Clark. *Natural-born cyborgs: Minds, technologies, and the future of human intelligence*. Oxford University Press, 2004.

[69] R.A. Clark, Y. Pua, K. Fortin, C. Ritchie, K.E. Webster, L. Denehy, and A.L. Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & posture*, 36(3):372–377, 2012.

[70] C. Colwell, H. Petrie, D. Kornbrot, A. Hardwick, and S. Furner. Haptic virtual reality for blind computer users. In *Proceedings of the Third International ACM Conference on Assistive Technologies*, Assets '98, pages 92–99, New York, NY, USA, 1998. ACM.

[71] S. Cooper, A. Hertzmann, and Z. Popović. Active learning for real-time motion controllers. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[72] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proc. of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 135–142, New York, NY, USA, 1993. ACM.

[73] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992.

[74] J.E. Cutting and L.T. Kozlowski. Recognizing Friends By Their Walk - Gait Perception Without Familiarity Cues. *Bulletin Of The Psychonomic Society*, 9(5):353–356, 1977.

[75] A. Cypher and D.C. Halbert. *Watch what I do: programming by demonstration*. MIT press, 1993.

[76] J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components; a biologically plausible computational model. *Imitation in animals and artifacts*, page 327, 2002.

[77] J.E. Deutsch, J. Latonio, G.C. Burdea, and R. Boian. Post-stroke rehabilitation with the rutgers ankle system: A case study. *Presence: Teleoper. Virtual Environ.*, 10(4):416–430, August 2001.

[78] P. Di Giuseppantonio Di Franco, C. Camporesi, F. Galeazzi, and M. Kallmann. 3d printing and immersive visualization for improved perception and interaction with past material culture. *Presence: Teleoperators and Virtual Environments*, in printing, 2015.

[79] P. Di Giuseppantonio Di Franco and F. Galeazzi. Western han dynasty mural tombs:from the use of integrated technologies to the cybermap. In *Proceedings of the 38th Annual Conference on Computer Applications and Quantitative Methods in Archaeology,*, volume 2494 of *CAA 2010*. BAR International, 2013.

[80] P. Di Giuseppantonio Di Franco, F. Galeazzi, and C. Camporesi. 3d virtual dig: a 3d application for teaching fieldwork in archaeology. *In Internet Archaeology*, 32, 2012.

[81] Digia. *Qt Project*, 2014 (accessed September 20, 2014). *http://qt-project.org*.

[82] K.U. Doerr and F. Kuester. Cglx: A scalable, high-performance visualization framework for networked display environments. *Visualization and Computer Graphics, IEEE Transactions on*, 17(3):320–332, March 2011.

[83] M. Dontcheva, G. Yngve, and Z. Popović. Layered acting for character animation. *ACM Transactions on Graphics*, 22(3):409–416, 2003.

[84] R. Doonan and M. Boyd. Contact: digital modelling of object and process in artefact teaching. *Touch in Museums: policy and practice in object handling*, pages 107–120, 2008.

[85] S. Dudley. *Museum materialities: Objects, engagements, interpretations*. Routledge, 2013.

[86] P.S. Dukes, A. Hayes, L.F. Hodges, and M. Woodbury. Punching ducks for post-stroke neurorehabilitation: System design and initial exploratory feasibility study. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 47–54, March 2013.

[87] S. Eilemann, M. Makhinya, and R. Pajarola. Equalizer: A scalable parallel rendering framework. *Visualization and Computer Graphics, IEEE Transactions on*, 15(3):436 –452, may-june 2009.

[88] FLTK. *FLTK: Fast Light Toolkit*, 2014 (accessed September 20, 2014). *http://www.fltk.org*.

[89] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii. inform: dynamic physical affordances and constraints through shape and object actuation. In *UIST*, pages 417–426, 2013.

[90] M. Forte. *La villa di Livia: un percorso di ricerca di archeologia virtuale*, volume 41. L'Erma di Bretschneider, 2007.

[91] J. Fox and J.N. Bailenson. Virtual self-modeling: The effects of vicarious reinforcement and identification on exercise behaviors. *Media Psychology*, 12(1):1–25, 2009.

[92] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster. Full body gait analysis with kinect. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1964–1967. IEEE, 2012.

[93] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester. Accuracy of the microsoft kinect sensor for measuring movement in people with parkinson's disease. *Gait & Posture*, 39(4):1062 – 1068, 2014.

[94] J. Gascón, J.M. Bayona, J.M Espadero, and M.A. Otaduy. Blendercave: Easy vr authoring for multi-screen displays. In *Proceedings of the Ibero-American Symposium in Computer Graphics*, SIACG 2011, Otaduy, Spain, 2011. Georgia Institute of Technology.

[95] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[96] P. Gebhard, M. Kipp, M. Klesen, and T. Rist. What are they going to talk about? towards life-like characters that reflect on interactions with users. In *Proc. of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE'03)*, 2003.

[97] L. Geurts, V. Vanden Abeele, J. Husson, F. Windey, M. Van Overveldt, J. Annema, and S. Desmet. Digital games for physical therapy: fulfilling the need for calibration and adaptation. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 117–124. ACM, 2011.

[98] J.J. Gibson. The ecological approach to visual perception. *Houghton Mifflin*, 1979.

[99] B. Gilliam. The perception of spatial layout from static optical information. In *Perception of Space and Motion*. W. Epstein and S. Rogers, eds. - Academic Press, Inc, 1995.

[100] P. Glardon, R. Boulic, and D. Thalmann. D.: A coherent locomotion engine extrapolating beyond experimental data. In *Proceedings of Computer Animation and Social Agent*, pages 73–84, 2004.

[101] gl.tter. *WiiYourself!*, 2014 (accessed September 20, 2014). *http://wiiyourself.gl.tter.org*.

[102] S. Goldin-Meadow. *Hearing gesture: How our hands help us think*. Harvard University Press, 2005.

[103] M. R. Golomb, B. C. McDonald, S. J. Warden, J. Yonkman, A. J. Saykin, B. Shirley, M. Huber, B. Rabin, M. Abdelbaky, M. E. Nwosu, M. Barkat-Masih, and G. C. Burdea. In-home virtual reality videogame telerehabilitation in adolescents with hemiplegic cerebral palsy. *Archives of physical medicine and rehabilitation*, 91(1):1–8, Jan 2010.

[104] M.R. Golomb, M. Barkat-Masih, B. Rabin, M. AbdelBaky, M. Huber, and G. Burdea. Eleven months of home virtual reality telerehabilitation - lessons learned. In *Virtual Rehabilitation International Conference, 2009*, pages 23–28, June 2009.

[105] F.S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, March 1998.

[106] M. Grealy and B. Nasser. The use of virtual reality in assisting rehabilitation. *Advances in Clinical Neuroscience and Rehabilitation*, 13(9):19–20, 2013.

[107] A.D. Gregory, S.A. Ehmann, and M.C. Lin. intouch: interactive multiresolution modeling and 3d painting with a haptic interface. In *Virtual Reality, 2000. Proceedings. IEEE*, pages 45–52, 2000.

[108] DavidB. Grimes and RajeshP.N. Rao. Learning actions through imitation and exploration: Towards humanoid robots that learn from humans. In B. Sendhoff, E. Krner, O. Sporns, H. Ritter, and K. Doya, editors, *Creating Brain-Like Intelligence*, volume 5436 of *Lecture Notes in Computer Science*, pages 103–138. Springer Berlin Heidelberg, 2009.

[109] Grinning Lizard. *Tiny XML*, 2014 (accessed September 20, 2014). *http://www.grinninglizard.com/tinyxml*.

[110] K. Grochow, S.L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, August 2004.

[111] R.E. Guadagno, J. Blascovich, J.N. Bailenson, and C. Mccall. Virtual humans and persuasion: The effects of agency and behavioral realism. *Media Psychology*, 10(1):1–22, 2007.

[112] A. Gupta and M.K. OMalley. *Robotic Exoskeletons for Upper Extremity Rehabilitation*, pages 371–396. I-Tech Education and Publishing, Vienna, Austria, 2007.

[113] M. Gutiérrez, P. Lemoine, D. Thalmann, and F. Vexo. Telerehabilitation: Controlling haptic virtual environments through handheld interfaces. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '04, pages 195–200, New York, NY, USA, 2004. ACM.

[114] M. Gutiérrez, F. Vexo, and D. Thalmann. *Stepping into Virtual Reality*. Springer-Verlag TELOS, Santa Clara, CA, USA, 1 edition, 2008.

[115] J.J. Han, G. Kurillo, R.T. Abresch, E. de Bie, A. Nicorici Lewis, and R. Bajcsy. Upper extremity 3d reachable workspace analysis in dystrophinopathy using kinect. *Muscle & nerve*, 2015.

[116] Reflexion Health, 2014 (accessed September 20, 2014). *http://www.reflexionhealth.com*.

[117] R. Heck and M. Gleicher. Parametric motion graphs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, pages 129–136, New York, NY, USA, 2007. ACM.

[118] C. Hecker, B. Raabe, R.W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. Graph.*, 27(3):27:1–27:11, August 2008.

[119] E. Heineken and F. P. Schulte. Seeing size and feeling weight: The size-weight illusion in natural and virtual reality. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(1):136–144, 2007.

[120] B. Herbelin, P. Benzaki, F. Riquier, O. Renault, and D. Thalmann. Using physiological measures for emotional assessment: a computer-aided tool for cognitive and behavioural therapy. In *ICDVRAT 2004*, VRLAB-CONF-2007-010, pages 307–314, 2004.

[121] H.E. Hershfield, D.G. Goldstein, W.F. Sharpe, J. Fox, L. Yeykelis, L.L. Carstensen, and J.N. Bailenson. Increasing saving behavior through age-progressed renderings of the future self. *Journal of Marketing Research*, 48(SPL):S23–S37, 2011.

[122] E.S.L. Ho, J.C.P. Chan, T. Komura, and H. Leung. Interactive partner control in close interactions for real-time applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 9(3):21, 2013.

[123] E.S.L. Ho and T. Komura. Character motion synthesis by topology coordinates. *Computer Graphics Forum*, 28:299–308, 2009.

[124] E.S.L. Ho, T. Komura, and C. Tai. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.*, 29(4):33:1–33:8, July 2010.

[125] H.G. Hoffmann. Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 59–63. IEEE, 1998.

[126] M.K. Holden. Virtual environments for motor rehabilitation: review. *Cyberpsichology and Behavior*, 8(3):187–211, 2005.

[127] M.K. Holden, T.A. Dyar, and E. Schwamm, L.and Bizzi. Virtual-environment-based telerehabilitation in patients with stroke. *Presence: Teleoperators and Virtual Environments*, 14(2):214–233, April 2005.

[128] M.K. Holden, E. Todorov, J. Callahan, and E. Bizzi. Virtual environment training improves motor performance in two patients with stroke: case report. *Journal of Neurologic Physical Therapy*, 23(2):57–67, 1999.

[129] E.J.S. Hovenga. *Health Informatics: An Overview*. Studies in health technology and informatics. IOS Press, 2010.

[130] Y. Huang, C. Camporesi, and M. Kallmann. Immersive interfaces for building parameterized motion databases. In *Proceedings of the 12th International Conference on Intelligent Virtual Agents*, IVA'12, pages 474–476, Berlin, Heidelberg, 2012. Springer-Verlag.

[131] Y. Huang and M. Kallmann. Interactive demonstration of pointing gestures for virtual trainers. In *Proceedings of 13th International Conference on Human-Computer Interaction*, San Diego, CA, 2009.

[132] Y. Huang and M. Kallmann. Motion parameterization with inverse blending. In *Proceedings of the Third International Conference on Motion in Games*, MIG'10, pages 242–253, Berlin, Heidelberg, 2010. Springer-Verlag.

[133] Y. Huang and M. Kallmann. Planning motions for virtual demonstrators. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents*, volume 8637 of *Lecture Notes in Computer Science*, pages 190–203. Springer International Publishing, 2014.

[134] Y. Huang, M. Mahmudi, and M. Kallmann. Planning humanlike actions in blending spaces. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2653–2659, Sept 2011.

[135] J. Hummel, J. Dodiya, R. Wolff, A. Gerndt, and T. Kuhlen. An evaluation of two simple methods for representing heaviness in immersive virtual environments. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 87–94. IEEE, 2013.

[136] G. Humphreys, I. Buck, M. Eldridge, and P. Hanrahan. Distributed rendering for scalable displays. In *Supercomputing, ACM/IEEE 2000 Conference*, pages 30–30, Nov 2000.

[137] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P.D. Kirchner, and J.T. Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Trans. Graph.*, 21(3):693–702, July 2002.

[138] E. Hutchins. Material anchors for conceptual blends. *Journal of pragmatics*, 37(10):1555–1577, 2005.

[139] iCinema: centre for Interactive Cinema Research. iCASTS - MiningVR, 2014 (accessed September 20, 2014). *http://www.icinema.unsw.edu.au*.

[140] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1398–1403, 2002.

[141] i'm in VR. Middle VR: Unity VR Plug-In, 2014 (accessed September 20, 2014). *http://www.imin.fr/middlevr*.

[142] InnaLabs. InnaLabs miniAHRS m2 Users Manual, 2014 (accessed September 20, 2014). *http://www.innalabs.com*.

[143] National Instruments. Introduction to Distributed Clock Synchronization and the IEEE 1588 Precision Time Protocol, 2014 (accessed September 20, 2014). *http://www.ni.com/white-paper/2822/en/*.

[144] D. Jack, R. Boian, A. Merians, S.V. Adamovich, M. Tremaine, M. Recce, G.C. Burdea, and H. Poizner. A virtual reality-based exercise program for stroke rehabilitation. In *Proceedings of the Fourth International ACM Conference on*

Assistive Technologies, Assets '00, pages 56–63, New York, NY, USA, 2000. ACM.

[145] J. Jacobson, M. Le Renard, J. Lugrin, and M. Cavazza. The caveut system: immersive entertainment based on a game engine. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ACE '05, pages 184–187, New York, NY, USA, 2005. ACM.

[146] J. Jacobson and M. Lewis. Game engine virtual reality with caveut. *Computer*, 38(4):79–82, April 2005.

[147] G. Jansson. Can a haptic force feedback display provide visually impaired people with useful information about texture roughness and 3d form of virtual objects? In *Proceedings of the 2nd European conference on disability, virtual reality, and associated technologies (ECDVRAT)*, pages 105–111, 1998.

[148] B. Jeong, L. Renambot, R. Jagodic, R. Singhm, J. Aguilera, A. Johnson, and J. Leigh. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *SC 2006 Conference, Proceedings of the ACM/IEEE*, pages 24–24, Nov 2006.

[149] W. L. Johnson, J. W. Rickel, and J. C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial intelligence in education*, 11(1):47–78, 2000.

[150] S. Jörg, A. Normoyle, and A. Safonova. How responsiveness affects players' perception in digital games. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '12, pages 33–38, New York, NY, USA, 2012. ACM.

[151] A. Juarez, B. Schonenberg, and C. Bartneck. Implementing a low-cost cave system using the cryengine2. *Entertainment Computing*, 1(3-4):157–164, 2010.

[152] Y. Jung, S. Yeh, and J. Stewart. Tailoring virtual reality technology for stroke rehabilitation: A human factors design. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 929–934, New York, NY, USA, 2006. ACM.

[153] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue. Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In *Fourth Int. Workshop on Algorithmic Foundations on Robotics*, 2000.

[154] M. Kallmann. Analytical inverse kinematics with body posture control. *Computer Animation and Virtual Worlds (CAVW)*, 19(2):79–91, 2008.

[155] M. Kallmann. *GraphSim*, 2014 (accessed September 20, 2014). *http://graphics.ucmerced.edu/software.html*.

[156] C. Kayalar and S. Balcısoy. Augmented reality based user interfaces to assist fieldwork on excavation sites. *On the Road to Reconstructing the Past. Computer Applications and Quantitative Methods in Archaeology (CAA). Proceedings of the 36th International Conference*, pages 307–311, 2008.

[157] S. Kenderdine, C. Camporesi, and M. Forte. Rhizome of western han: an omnispatial theatre for archaeology. In *Computer Applications and Quantitative Methods in Archaeology (CAA). Proceedings of the 39th International Conference*, Amsterdam, 2011. Pallas Publications.

[158] E. A. Keshner. Virtual reality and physical rehabilitation: a new toy or a new research and rehabilitation tool? *J Neuroengineer Rehabilitation*, 1(1):8, Dec 2004.

[159] T. Kielmann, H. E. Bal, S. Gorlatch, K. Verstoep, and R. F.H. Hofman. Network performance-aware collective communication for clustered wide-area systems. *Parallel Computing*, 27(11):1431–1456, 2001.

[160] G. Kim. *Designing Virtual Reality Systems: The Structured Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[161] D. Kirsh. Problem solving and situated cognition. *In Robbins, P., and Aydede, M. (Eds.), The Cambridge Handbook of Situated Cognition*, pages 264–307, 2009.

[162] D. Kirsh. Comparing tangible and virtual exploration of archaeological objects. *In Forte, M. (Ed.), Cyber-Archaeology*, 2177:119–124, 2010.

[163] D. Kirsh. Thinking with external representations. In *Cognition Beyond the Brain*, pages 171–194. Springer, 2013.

[164] R. Kizony, N. Katz, et al. Adapting an immersive virtual reality system for rehabilitation. *The Journal of Visualization and Computer Animation*, 14(5):261–268, 2003.

[165] R. Kizony, P.L. Weiss, Y. Feldman, M. Shani, O. Elion, R. Kizony, P.L. Weiss, R. Kizony, S. Harel, and I. Baum-Cohen. Evaluation of a tele-health system for upper extremity stroke rehabilitation. In *Virtual Rehabilitation (ICVR), 2013 International Conference on*, pages 80–86, Aug 2013.

[166] R. L. Klatzky, S. J. Lederman, and V. A. Metzger. Identifying objects by touch: An expert system. *Perception & Psychophysics*, 37(4):299–302, 1985.

[167] J.M. Knapp and J.M. Loomis. Limited field of view of head-mounted displays is not the cause of distance underestimation in virtual environments. *Presence: Teleoperators and Virtual Environments*, 13(5):572–577, October 2004.

[168] B.W. Knerr, D.R. Lampton, M. Thomas, B.D. Corner, and J.R. Grosse. Virtual environments for dismounted soldier simulation, training, and mission rehearsal: Results of the fy 2002 culminating event. Technical report, DTIC Document, 2003.

[169] S. Kopp and I. Wachsmuth. Model-based animation of co-verbal gesture. *Computer Animation, 2002. Proceedings of*, pages 252–257, 2002.

[170] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 559–568, 2004.

[171] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Trans. Graph.*, 21(3):473–482, 2002.

[172] L.T. Kozlowski and J.E. Cutting. Recognizing the gender of walkers from point-lights mounted on ankles: Some second thoughts. *Perception & Psychophysics*, 23(5):459–459, 1978.

[173] R. M. Krauss. Why do we gesture when we speak? *Current Directions in Psychological Science*, 7:54–60, 1998.

[174] O. Kreylos. Environment-independent vr development. In *Proc. of the 4th International Symposium on Advances in Visual Computing*, ISVC '08, pages 901–912, Berlin, Heidelberg, 2008. Springer-Verlag.

[175] M. W. Krueger, T. Gionfriddo, and K. Hinrichsen. Videoplacean artificial reality. In *ACM SIGCHI Bulletin*, volume 16, pages 35–40. ACM, 1985.

[176] R. Kuck, J. Wind, K. Riege, and M. Bogen. Improving the avango vr/ar framework: Lessons learned. In *5th Workshop of the GI-VR/AR Group*, pages 209–220. Shaker, 2008.

[177] G. Kurillo, T. Koritnik, T. Bajd, and R. Bajcsy. Real-time 3d avatars for tele-rehabilitation in virtual reality. *Studies in health technology and informatics*, 163:290296, 2011.

[178] J.C.K. Lai, J. Woo, E. Hui, and W.M. Chan. Telerehabilitationa new model for community-based stroke rehabilitation. *Journal of telemedicine and telecare*, 10(4):199–205, 2004.

[179] G. Lakoff and M. Johnson. *Philosophy in the flesh: The embodied mind and its challenge to western thought*. Basic books, 1999.

[180] B. Lange, C. Chang, E. Suma, B. Newman, A.S. Rizzo, and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1831–1834. IEEE, 2011.

[181] B. Lange, S. M. Flynn, and A.S. Rizzo. Game-based telerehabilitation. *European journal of physical and rehabilitation medicine*, 45(1):143–151, Mar 2009.

[182] B. Lange, S. Koenig, C. Chang, E. McConnell, E. Suma, M. Bolas, and A.S. Rizzo. Designing informed game-based rehabilitation tasks leveraging advances in virtual reality. *Disability and rehabilitation*, 34(22):1863–1870, 2012.

[183] B. Lange, S. Koenig, E. McConnell, C. Chang, R. Juang, E. Suma, M. Bolas, and A.S. Rizzo. Interactive game-based rehabilitation using the microsoft kinect. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 171–172. IEEE, 2012.

[184] J. Lanier. Virtually there. *Scientific American*, 284(4):66–75, 2001.

[185] T. Le Naour, N. Courty, and S. Gibet. Spatiotemporal coupling with the 3d+ t motion laplacian. *Computer Animation and Virtual Worlds*, 24(3-4):419–428, 2013.

[186] Leap. *Leap Motion*, 2014 (accessed September 20, 2014). *https://www.leapmotion.com*.

[187] R.S. Leder, G. Azcarate, R. Savage, S. Savage, L.E. Sucar, D. Reinkensmeyer, C. Toxtli, E. Roth, and A. Molina. Nintendo wii remote for computer simulated arm and wrist therapy in stroke survivors with upper extremity hemipariesis. In *Virtual Rehabilitation, 2008*, page 74, 2008.

[188] S. J. Lederman and L. A. Jones. Tactile and haptic illusions. *Haptics, IEEE Transactions on*, 4(4):273–294, 2011.

[189] S. J. Lederman and R. L. Klatzky. Haptic classification of common objects: Knowledge-driven exploration. *Cognitive psychology*, 22(4):421–459, 1990.

[190] D. Lee, M. Lim, S. Han, and K. Lee. Atlas: A scalable network framework for distributed virtual environments. *Presence*, 16(2):125–156, April 2007.

[191] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, 2002.

[192] N. Lercari. Nuove forme di comunicazione per nu.m.e. In Francesca Bocchi and Rosa Smurra, editors, *La Storia della Cittá per il Museo Virtuale di Bologna. Un decennio di Ricerche nel Dottorato di Storia e Informatica*, pages 217–225. Bononia University Press, 2010.

[193] D.E. Levac and J. Galvin. When is virtual reality therapy? *Archives of Physical Medicine and Rehabilitation*, 94(4):795–798, 2013.

[194] A.I. Levine, S. DeMaria, A.D. Schwartz, and A.J. Sim. *The Comprehensive Textbook of Healthcare Simulation*. Springer, 2013.

[195] S. Levine, J.M. Wang, A. Haraux, Z. Popović, and V. Koltun. Continuous character control with low-dimensional embeddings. *ACM Trans. Graph.*, 31(4):28:1–28:10, July 2012.

[196] M. Lewis and J. Jacobson. Game engines in scientific research. *Commun. ACM*, 45(1):27–31, January 2002.

[197] P.P. Li, S. Whitman, R. Mendoza, and J. Tsiao. Parvox-a parallel splatting volume rendering system for distributed visualization. In *Parallel Rendering, 1997. PRS 97. Proceedings. IEEE Symposium on*, pages 7–14, 113–14, Oct 1997.

[198] C.J. Lin, T. Sun, H. Chen, and P. Cheng. Evaluation of visually-controlled task performance in three dimension virtual reality environment. In Randall Shumaker, editor, *Virtual and Mixed Reality*, volume 5622 of *Lecture Notes in Computer Science*, pages 465–471. Springer Berlin Heidelberg, 2009.

[199] J.J.. Lin, H.B. Duh, D.E. Parker, H. Abi-Rached, and T.A. Furness. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 164–171, 2002.

[200] P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. A. Turner. Real-time, continuous level of detail rendering of height fields. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 109–118, New York, NY, USA, 1996. ACM.

[201] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: Applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, December 2003.

[202] K.C. Liu and Z. Popović. Synthesis of complex dynamic character motion from simple animations. *ACM Trans. Graph.*, 21(3):408–416, July 2002.

[203] S. Lombeyda, L. Moll, M. Shand, D. Breen, and A. Heirich. Scalable interactive volume rendering using off-the-shelf components. In *Parallel and Large-Data Visualization and Graphics, 2001. Proceedings. IEEE 2001 Symposium on*, pages 115–158, Oct 2001.

[204] J.M. Loomis, J.A. Da Silva, J.W. Philbeck, and S.S. Fukusima. Visual perception of location and distance. *Current Directions in Psychological Science*, 5:72–77, 1996.

[205] C. Loscos, F. Tecchia, A. Frisoli, M. Carrozzino, H. R. Widenfeld, D. Swapp, and M. Bergamasco. The museum of pure form: touching real statues in an immersive virtual museum. In *Proceedings of the 5th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pages 271–279. Eurographics Association, 2004.

[206] L.P. Lowes, L.N. Alfano, B.A. Yetter, L. Worthen-Chaudhari, W. Hinchman, J. Savage, P. Samona, K.M. Flanigan, and J.R. Mendell. Proof of concept of the ability of the kinect to quantify upper extremity function in dystrophinopathy. *PLoS currents*, 5, 2013.

[207] H. Lü and Y. Li. Gesture coder: a tool for programming multi-touch gestures by demonstration. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 2875–2884. ACM, 2012.

[208] J. Lugrin, F. Charles, M. Cavazza, M. Le Renard, J. Freeman, and J. Lessiter. Caveudk: A vr game engine middleware. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, VRST '12, pages 137–144, New York, NY, USA, 2012. ACM.

[209] W. Ma, S. Xia, J.K. Hodgins, X. Yang, C. Li, and Z. Wang. Modeling style and variation in human motion. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, 2010.

[210] G. MacGregor. Making sense of the past in the present: a sensory analysis of carved stone balls. *World archaeology*, 31(2):258–271, 1999.

[211] N. Magnenat-Thalmann and G. Papagiannakis. Virtual worlds and augmented reality in cultural heritage applications. *In Baltsavias, M., Gruen, A., Van Gool, L., Pateraki, M. (Eds.), Recording, Modeling, and Visualization of Cultural Heritage: Proceedings of the International Workshop*, pages 419–430, 2005.

[212] M. Mahmudi and M. Kallmann. Feature-based locomotion with inverse branch kinematics. In *Proceedings of the 4th International Conference on Motion in Games*, MIG'11, pages 39–50, Berlin, Heidelberg, 2011. Springer-Verlag.

[213] R. Maiocchi. 3-d character animation using motion capture. In Nadia Magnenat Thalmann and Daniel Thalmann, editors, *Interactive Computer Animation*, pages 10–39, Upper Saddle River, NJ, USA, 1996. Prentice-Hall, Inc.

[214] T. H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Chicago, IL, 1994.

[215] T. Matlock, D. Sparks, J. L. Matthews, J. Hunter, and S. Huette. Smashing new results on aspectual framing: How people talk about car accidents. *Studies in Language*, 36(3):699–720, 2012.

[216] R. McDonnell, M. Breidt, and H.H. Bülthoff. Render me real?: investigating the effect of render style on the perception of animated virtual humans. *ACM Trans. Graph.*, 31(4):91:1–91:11, July 2012.

[217] R. McDonnell, S. Jörg, J. McHugh, F.N. Newell, and C. O'Sullivan. Investigating the role of body shape on the perception of emotion. *ACM Transations on Applied Perception*, 6(3):14:1–14:11, September 2009.

[218] R.P. McMahan, D.A. Bowman, D.J. Zielinski, and R.B. Brady. Evaluating display fidelity and interaction fidelity in a virtual reality game. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):626–633, April 2012.

[219] R.P. McMahan, D. Gorton, J. Gresock, W. McConnell, and D.A. Bowman. Separating the effects of level of immersion and 3d interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '06, pages 108–111, New York, NY, USA, 2006. ACM.

[220] D. McNeill. *Gesture and thought*. University of Chicago Press, 2008.

[221] A.S. Merians, D. Jack, R. Boian, M. Tremaine, G.C. Burdea, S.V. Adamovich, M. Recce, and H. Poizner. Virtual reality–augmented rehabilitation for patients following stroke. *Journal of Physical Therapy*, 82(9):898–915, 2002.

[222] D. Michael, N. Pelekanos, I. Chrysanthou, P. Zaharias, L. L. Hadjigavriel, and Y. Chrysanthou. Comparative study of interactive systems in a museum. In *Digital Heritage*, pages 250–261. Springer, 2010.

[223] D.R. Michael and S.L. Chen. *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.

[224] Microsoft. *Microsoft Kinect SDK*, 2014 (accessed September 20, 2014). *http://www.microsoft.com/en-us/kinectforwindows*.

[225] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for Industrial Applications*, pages 282–292. International Society for Optics and Photonics, 1995.

[226] J. Min, H. Liu, and J. Chai. Synthesis and editing of personalized stylistic human motion. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, pages 39–46, New York, NY, USA, 2010. ACM.

[227] M. Minsky, O. Ming, O. Steele, F. P. Brooks Jr, and M. Behensky. Feeling and seeing: issues in force display. *ACM SIGGRAPH Computer Graphics*, 24(2):235–241, 1990.

[228] A. Mobini, S. Behzadipour, and M. Saadat Foumani. Accuracy of Kinect's skeleton tracking for upper body rehabilitation applications. *Disabil Rehabil Assist Technol*, 9(4):344–352, Jul 2014.

[229] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, November 2006.

[230] S. Molnar, J. Eyles, and J. Poulton. Pixelflow: High-speed rendering using image composition. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '92, pages 231–240, New York, NY, USA, 1992. ACM.

[231] G. Monheit and N.I. Badler. A kinematic model of the human spine and torso. *IEEE Comput. Graph. Appl.*, 11(2):29–38, 1991.

[232] C. Mueller. The sort-first rendering architecture for high-performance graphics. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, pages 75–ff., New York, NY, USA, 1995. ACM.

[233] C. Mueller. Hierarchical graphics databases in sort-first. In *Parallel Rendering, 1997. PRS 97. Proceedings. IEEE Symposium on*, pages 49–57, 117, Oct 1997.

[234] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. *ACM Trans. Graph.*, 24(3):1062–1070, July 2005.

[235] S. Muraki, M. Ogata, K. Ma, K. Koshizuka, K. Kajihara, X. Liu, Y. Nagano, and K. Shimokawa. Next-generation visual supercomputing using pc clusters with volume graphics hardware devices. In *Proceedings of the 2001 ACM/IEEE Conference on Supercomputing*, SC '01, pages 51–51, New York, NY, USA, 2001. ACM.

[236] A. Naceri, R. Chellali, F. Dionnet, and S. Toma. Depth perception within virtual environments: A comparative study between wide screen stereoscopic displays and head mounted devices. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, pages 460–466, 2009.

[237] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S.n Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.

[238] M. Narayan, L. Waugh, X. Zhang, P. Bafna, and D. Bowman. Quantifying the benefits of immersion for collaboration in virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '05, pages 78–81, New York, NY, USA, 2005. ACM.

[239] R. Navaratnam, AW. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[240] M. Neff, M. Kipp, I. Albrecht, and H. Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics*, 27(1):5:1–5:24, March 2008.

[241] U. Nehmzow, O. Akanyeti, C. Weinrich, T. Kyriacou, and S.A. Billings. Robot programming by demonstration through system identification. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.*, pages 801–806. IEEE, 2007.

[242] C.G. Nevill-Manning. Programming by demonstration. *New Zealand Journal of Computing*, 4(2):15–24, 1993.

[243] A.Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, pages 363–372. Springer, 2006.

[244] T. Ni, D.A. Bowman, and J. Chen. Increased display size and resolution improve task performance in information-rich virtual environments. In *Proceedings of Graphics Interface 2006*, GI '06, pages 139–146, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[245] T. Ni, G. S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball, and R. May. A survey of large high-resolution display technologies, techniques, and applications. In *Proceedings of the IEEE conference on Virtual Reality*, VR '06, pages 223–236, Washington, DC, USA, 2006. IEEE Computer Society.

[246] P.H. Nirnimesh and P.J. Narayanan. Garuda: A scalable tiled display wall using commodity pcs. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5):864 –877, sept.-oct. 2007.

[247] M. Nixon, Y. Chen, and A. Howard. Quantitative evaluation of the microsoft kinect for use in an upper extremity virtual rehabilitation environment. In

International Conference on Virtual Rehabilitation (ICVR), Philadelphia, PA, U.S.A., May 2013.

[248] S.S. Noh, S.D. Hong, and J.W. Park. Using a game engine technique to produce 3d entertainment contents. In *Proceedings of the 16th International Conference on Artificial Reality and Telexistence–Workshops*, ICAT '06, pages 246–251, Washington, DC, USA, 2006. IEEE Computer Society.

[249] T. Noma, L. Zhao, and N.I. Badler. Design of a virtual human presenter. *IEEE Computer Graphics and Applications*, 20(4):79–85, 2000.

[250] C. Norkin. *Measurement of Joint Motion. A Guide to Goniometry.* F.A. Davis Company, Philadelphia, PA, 2003.

[251] D. A. Norman. Les artefacts cognitifs. *In Conein, B., Dodier, N., & Thvenot, L. (Eds.), Raisons pratiques n.4. Les objets dans laction*, 4(1993):15–34, 1993.

[252] NVidia. NVidia 3D vision, 2014 (accessed September 20, 2014). *http://www.nvidia.com/object/3d-vision-main.html*.

[253] NVidia. NVidia FrameLock, 2014 (accessed September 20, 2014). *http://www.nvidia.com/object/IO_10794.html*.

[254] NVidia. NVidia GenLock, 2014 (accessed September 20, 2014). *http://www.nvidia.com/object/IO_10793.html*.

[255] NVidia. NVidia Mosaic, 2014 (accessed September 20, 2014). *http://www.nvidia.com/object/nvidia-mosaic-technology.html*.

[256] S. Obdrzalek, G. Kurillo, J. Han, T. Abresch, and R. Bajcsy. Real-time human pose detection and tracking for tele-rehabilitation in virtual reality. *Stud Health Technol Inform*, 173, 2012.

[257] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1188–1193, Aug 2012.

[258] Oculus VR. *Oculus Rift*, 2014 (accessed September 20, 2014). *http://www.oculusvr.com*.

[259] International Society of Presence Research (ISPR). *What is Presence?*, 2014 (accessed August 20, 2014). *http://ispr.info*.

[260] E. C. Olson. Cluster juggler - pc cluster virtual reality - m.sc. thesis, 2002.

[261] L. Omelina, B. Jansen, B. Bonnechre, S. Van Sint Jan, and J. Cornelis. Serious games for physical rehabilitation: designing highly configurable and adaptable games. In *Proc 9th Intl Conf. Disability, Virutal Reality & Associated Technologies, Laval, France*, 2012.

[262] VideoLan Organization. *VideoLan library*, 2014 (accessed September 20, 2014). *http://www.videolan.org*.

[263] R.A. Pavlik and J.M. Vance. Vr jugglua: A framework for vr applications combining lua, openscenegraph, and vr juggler. In *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2012 5th Workshop on*, pages 29–35, March 2012.

[264] M. Peinado, D. Meziat, D. Maupu, D. Raunhardt, D. Thalmann, and R. Boulic. Full-Body Avatar Control with Environment Awareness. *IEEE Computer Graphics And Applications*, 29:62–75, 2009.

[265] D. Perez-Marcos, M. Solazzi, W. Steptoe, O. Oyekoya, A. Frisoli, T. Weyrich, A. Steed, F. Tecchia, M. Slater, and M. V. Sanchez-Vives. A fully immersive set-up for remote interaction and neurorehabilitation based on virtual body ownership. *Frontiers in Neurology*, 3:110, 2012.

[266] J.C. Perry, J. Andureu, F.I. Cavallaro, J. Veneman, S. Carmien, and T. Keller. Effective game use in neurorehabilitation: user-centered perspectives. *Handbook of Research on Improving Learning and Motivation through Educational Games, IGI Global*, 2010.

[267] L. Phillips, B. Ries, M. Kaeding, and V. Interrante. Avatar self-embodiment enhances distance perception accuracy in non-photorealistic immersive virtual environments. In *Proceedings of the 2010 IEEE Virtual Reality Conference*, VR '10, pages 115–1148, Washington, DC, USA, 2010. IEEE Computer Society.

[268] M. Piqueras, E. Marco, M. Coll, F. Escalada, A. Ballester, C. Cinca, R. Belmonte, and J. M. Muniesa. Effectiveness of an interactive virtual telerehabilitation system in patients after total knee arthoplasty: a randomized controlled trial. *Journal of Rehabilitation Medicine*, 45(4):392–396, Apr 2013.

[269] D. Poirier-Quinot, D. Touraine, and B.F.G. Katz. Blendercave: A multimodal scene graph editor for virtual reality. In *Proceedings of the International Conference on auditory Display*, ICaD 2013, ódź, Poland, 2013. Georgia Institute of Technology.

[270] N.S. Pollard and J.K. Hodgins. Generalizing demonstrated manipulation tasks. In *Algorithmic Foundations of Robotics V*, pages 523–540. Springer, 2004.

[271] V.G. Popescu, G.C. Burdea, M. Bouzit, and V.R. Hentz. A virtual-reality-based telerehabilitation system with force feedback. *Information Technology in Biomedicine, IEEE Transactions on*, 4(1):45–51, March 2000.

[272] R. Poppe. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst.*, 108(1-2):4–18, October 2007.

[273] B. Preim and D. Bartz. *Visualization in medicine: theory, algorithms, and applications*. Morgan Kaufmann, 2007.

[274] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2007.

[275] F.H. Previc. The neuropsychology of 3-D space. *Psychological Bulletin*, 124(2):123–164, Sep 1998.

[276] J. Psotka. Immersive training systems: Virtual reality and education and training. *Instructional Science*, 23(5-6):405–431, 1995.

[277] E.D. Ragan, R. Kopper, P. Schuchardt, and D.A. Bowman. Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):886–896, May 2013.

[278] P. Rajlich. Cave Quake II, 2014 (accessed September 20, 2014). *http://www.visbox.com/prajlich/caveQuake*.

[279] J. J. Ratey. *A user's guide to the brain: Perception, attention, and the four theatres of the brain*. Vintage Books, 2001.

[280] D. Raunhardt and R. Boulic. Motion constraint. *The Visual Computer*, 25(5-7):509–518, 2009.

[281] EON Reality. EON Icube, 2014 (accessed September 20, 2014). *http://www.eonreality.com*.

[282] P.S.A. Reitsma and N.S. Pollard. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM Transactions on Graphics*, 22(3):537–542, July 2003.

[283] Relievo. The Relievo Collection. Premium Replica of Van Gogh Masterpieces, 2014 (accessed September 20, 2014). *http://vangoghinternational.com*.

[284] C. Ren, L. Zhao, and A. Safonova. Human motion synthesis with optimization-based graphs. *Comput. Graph. Forum*, 29(2):545–554, 2010.

[285] S. Rhee, R. Ziegler, J. Park, M. Naef, and M. Gross. Low-cost telepresence for collaborative virtual environments. *in IEEE Transactions of Visualization and Computer Graphics (TVCG)*, 13:156–166, 2007.

[286] H. Rheingold. *Virtual Reality*. Simon & Schuster, Inc., New York, NY, USA, 1991.

[287] J. Rickel and W. L. Johnson. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13:343–382, 1998.

[288] U. Ritterfeld, M. Cody, and P. Vorderer. *Serious Games: Mechanisms and Effects*. Taylor & Francis, 2010.

[289] A.S. Rizzo, B.S. John, B. Newman, J. Williams, A. Hartholt, C. Lethin, and J.G. Buckwalter. Virtual reality as a tool for delivering PTSD exposure therapy and stress resilience training. *Military Behavioral Health*, 1:48–54, 2012.

[290] A. Robb, C. White, A. Cordar, A. Wendling, S. Lampotang, and B. Lok. A qualitative evaluation of behavior during conflict with an authoritative virtual human. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents*, volume 8637 of *Lecture Notes in Computer Science*, pages 397–409. Springer International Publishing, 2014.

[291] I. Rodriguez, M. Peinado, R. Boulic, and D. Meziat. Bringing the human arm reachable space to a virtual environment for its analysis. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, pages I–229. IEEE, 2003.

[292] D. Roetenberg, H. Luinge, and P. Slycke. Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. xsens motion technologies bv. Technical report, Xsens 3D Motion Tracking, 2009.

[293] D. Roetenberg, H. Luinge, and P. Veltink. Inertial and magnetic sensing of human movement near ferromagnetic materials. In *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 268–, Washington, DC, USA, 2003. IEEE Computer Society.

[294] M. Rønne Jakobsen and K. Hornbæk. Sizing up visualizations: Effects of display size in focus+context, overview+detail, and zooming interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1451–1460, New York, NY, USA, 2011. ACM.

[295] C.F. III Rose, M.F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, September 1998.

[296] C.F. III Rose, P.P.J. Sloan, and M.F. Cohen. Artist-directed inverse-kinematics using radial basis function interpolation. *Computational Graphics Forum*, 20(3):239–250, 2001.

[297] A.K. Roy, Y. Soni, and S. Dubey. Enhancing effectiveness of motor rehabilitation using kinect motion sensing technology. In *Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), 2013 IEEE*, pages 298–304. IEEE, 2013.

[298] A. Safonova and J.K. Hodgins. Construction and optimal search of interpolated motion graphs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 106, New York, NY, USA, 2007. ACM.

[299] M. Salvati, B. Le Callennec, and R. Boulic. A Generic Method for Geometric Contraints Detection. In *Eurographics*, 2004.

[300] R. Samanta, T. Funkhouser, and K. Li. Parallel rendering with k-way replication. In *Parallel and Large-Data Visualization and Graphics, 2001. Proceedings. IEEE 2001 Symposium on*, pages 75–153, Oct 2001.

[301] R. Samanta, T. Funkhouser, K. Li, and J.P. Singh. Hybrid sort-first and sort-last parallel rendering with a cluster of pcs. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, HWWS '00, pages 97–108, New York, NY, USA, 2000. ACM.

[302] R. Samanta, J. Zheng, T. Funkhouser, K. Li, and J. P. Singh. Load balancing for multi-projector rendering systems. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, HWWS '99, pages 107–116, New York, NY, USA, 1999. ACM.

[303] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[304] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *The Neuroscience of Social Interaction*, 1431:199–218, 2003.

[305] B. Schaeffer and C. Goudeseune. Syzygy: native pc cluster vr. In *Virtual Reality, 2003. Proceedings. IEEE*, pages 15 – 22, march 2003.

[306] C. Schönauer, T. Pintaric, and H. Kaufmann. Full body interaction for serious games in motor rehabilitation. In *Proceedings of the 2nd Augmented Human International Conference*, AH '11, pages 4:1–4:8, New York, NY, USA, 2011. ACM.

[307] T. Schou and H.J. Gardner. A wii remote, a game engine, five sensor bars and a virtual reality theatre. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, pages 231–234, New York, NY, USA, 2007. ACM.

[308] M.T. Schultheis and R.R. Mourant. Virtual reality and driving: The road to better assessment for cognitively impaired populations. *Presence: Teleoperators and Virtual Environments*, 10(4):431–439, 2001.

[309] N.E. Seymour, A.G. Gallagher, S.A. Roman, M.K. O'Brien, V.K. Bansal, D.K. Andersen, and R.M. Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery*, 236(4):458, 2002.

[310] SGI. SGI Onyx2, 2014 (accessed September 20, 2014). *http://www.sgi.com*.

[311] C. Shih, M. Chang, and C. Shih. A limb action detector enabling people with multiple disabilities to control environmental stimulation through limb action with a nintendo wii remote controller. *Research in Developmental Disabilities*, 31(5):1047 – 1053, 2010.

[312] C. Shih, M. Chang, and C. Shih. A new limb movement detector enabling people with multiple disabilities to control environmental stimulation through limb swing with a gyration air mouse. *Research in Developmental Disabilities*, 31(4):875 – 880, 2010.

[313] W. Shim and G. Kim. Designing for presence and performance: The case of the virtual fish tank. *Presence*, 12(4):374–386, Aug 2003.

[314] H.J. Shin, J. Lee, S.Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94, April 2001.

[315] T. Shiratori and J.K. Hodgins. Accelerometer-based user interfaces for the control of a physically simulated character. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 123:1–123:9, New York, NY, USA, 2008. ACM.

[316] T. Shiratori, H.S. Park, L. Sigal, Y. Sheikh, and J.K. Hodgins. Motion capture from body-mounted cameras. *ACM Trans. Graph.*, 30(4):31:1–31:10, July 2011.

[317] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, January 2013.

[318] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 1–8, June 2010.

[319] Sixense. *Stem System*, 2014 (accessed September 20, 2014). *http://sixense.com*.

[320] A. Skoglund, B. Iliev, and R. Palm. Programming-by-demonstration of reaching motions - a next-state-planner approach. *Robotics and Autonomous Systems*, 58(5):607–621, 2010.

[321] M. Slater, B. Spanlang, M. V. Sanchez-Vives, and O. Blanke. First person experience of body transfer in virtual reality. *PloS one*, 5(5):e10564, 2010.

[322] M. Slater and S. Wilbur. A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. *Presence Teleoperators and Virtual Environments*, 6(6):603–616, 1997.

[323] R. Slyper and J.K. Hodgins. Action capture with accelerometers. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 193–199, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[324] Jenkins Software. *RakNet - Multiplayer game network engine*, 2014 (accessed September 20, 2014). *http://www.jenkinssoftware.com*.

[325] PassMark Software. Videocard Benchmarks - NVidia Geforce 280 GTX vs NVidia Quadro 5000, 2014 (accessed September 20, 2014). *http://www.videocardbenchmark.net*.

[326] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[327] O. G. Staadt, J. Walker, C. Nuber, and B. Hamann. A survey and performance analysis of software platforms for interactive cluster-based multi-screen rendering. In *Proceedings of the Workshop on Virtual Environments 2003*, EGVE '03, pages 261–270, New York, NY, USA, 2003. ACM.

[328] W. Steptoe, A. Steed, and M. Slater. Human tails: Ownership and control of extended humanoid avatars. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):583–590, 2013.

[329] R.J. Sternberg. *Practical Intelligence in Everyday Life*. Cambridge University Press, 2000.

[330] G. Stoll, M. Eldridge, D. Patterson, A. Webb, S. Berman, R. Levy, C. Caywood, M. Taveira, S. Hunt, and P. Hanrahan. Lightning-2: A high-performance display subsystem for pc clusters. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 141–148, New York, NY, USA, 2001. ACM.

[331] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics*, 23(3):506–513, 2004.

[332] Structure. *Structure Mobile Sensor*, 2014 (accessed September 20, 2014). *http://structure.io*.

[333] L. A. Suchman. *Plans and situated actions: the problem of human-machine communication*. Cambridge university press, 1987.

[334] H.C. Sun and D.N. Metaxas. Automating gait generation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 261–270, New York, NY, USA, 2001. ACM.

[335] I. E. Sutherland. The ultimate display. *IFIP Congress*, pages 506–508, 1965.

[336] J.E. Swan II, A. Jones, E. Kolstad, M.A. Livingston, and H.S. Smallman. Egocentric depth judgments in optical, see-through augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):429–442, May 2007.

[337] W.R. Swartout. Lessons learned from virtual humans. *AI Magazine*, pages 9–20, 2010.

[338] J.D. Sweeney and R. Grupen. A model of shared grasp affordances from demonstration. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 27–35. IEEE, 2007.

[339] Synertial. IGS 180, 2014 (accessed September 20, 2014). *http://www.synertial.com*.

[340] Cyber Glove Systems. CyberGrasp force feedback system, 2014 (accessed September 20, 2014). *http://www.cyberglovesystems.com/products/cybergrasp/overview*.

[341] D.S. Tan, D. Gergle, P. Scupelli, and R. Pausch. With similar visual angles, larger displays improve spatial performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 217–224, New York, NY, USA, 2003. ACM.

[342] J. Tanenbaum, M.S. El-Nasr, and M. Nixon. *Nonverbal Communication in Virtual Worlds: Understanding and Designing Expressive Characters*. LULU Press, 2014.

[343] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H. Seidel, and B. Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30(3):18:1–18:12, May 2011.

[344] Bullet Team. *Bullet: Real-Time Physics Simulation*, 2014 (accessed September 20, 2014). *http://bulletphysics.org*.

[345] OGRE Team. *OGRE: Object-Oriented Graphics Rendering Engine*, 2014 (accessed September 20, 2014). *http://www.ogre3d.org*.

[346] R.J. Teather, A Pavlovych, W. Stuerzlinger, and IS. MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pages 43–50, March 2009.

[347] D. Thewlis, C. Bishop, N. Daniell, and G. Paul. Next generation low-cost motion capture systems can provide comparable spatial accuracy to high-end systems. *Journal of applied biomechanics*, 29(1):112–117, 2013.

[348] M. Thiebaux, A. Marshall, S. Marsella, and M. Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.

[349] Gergory T Thielman, Catherine M Dean, and AM Gentile. Rehabilitation of reaching after stroke: task-related training versus progressive resistive exercise. *Archives of physical medicine and rehabilitation*, 85(10):1613–1618, 2004.

[350] S.B. Thies, P. Tresadern, L. Kenney, D. Howard, J.Y. Goulermas, C Smith, and J Rigby. Comparison of linear accelerations from three measurement systems during reach & grasp. *Medical engineering & physics*, 29(9):967–972, 2007.

[351] W.B. Thompson, P. Willemsen, A.A. Gooch, S.H. Creem-Regehr, J.M. Loomis, and A.C. Beall. Does the quality of the computer graphics matter when judging distances in visually immersive environments. *Presence: Teleoperators and Virtual Environments*, 13(5):560–571, October 2004.

[352] L. P. Tost and M. Economou. Worth a thousand words? the usefulness of immersive virtual reality for learning in cultural heritage settings. *International Journal of Architectural Computing*, 7(1):157–176, 2009.

[353] D. Traum and J. Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*, AAMAS '02, pages 766–773, New York, NY, USA, 2002. ACM.

[354] D. Traum, W. Swartout, S. Marsella, and J. Gratch. Fight, flight, or negotiate: Believable strategies for conversing under crisis. In Themis Panayiotopoulos, Jonathan Gratch, Ruth Aylett, Daniel Ballin, Patrick Olivier, and Thomas Rist, editors, *Intelligent Virtual Agents*, volume 3661 of *Lecture Notes in Computer Science*, pages 52–64. Springer Berlin Heidelberg, 2005.

[355] Unity. Unity - Game Engine, 2014 (accessed September 20, 2014). *http://unity3d.com*.

[356] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 91–96, New York, NY, USA, 1995. ACM.

[357] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[358] T. Van der Schaaf, L. Renambot, D. Germans, H. Spoelder, and H. Bal. Retained mode parallel rendering for scalable tiled displays. In *Immersive Projection Technologies Symp.*, 2002.

[359] F. J. Varela, E. Rosch, and E. Thompson. *The embodied mind: Cognitive science and human experience*. MIT press, 1992.

[360] E. Velloso, A. Bulling, and H. Gellersen. Motionma: Motion modelling and analysis by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1309–1318, New York, NY, USA, 2013. ACM.

[361] VICON. *VICON Solutions*, 2014 (accessed September 20, 2014). *http://www.vicon.com*.

[362] Virtuix. *Virtuix Omni*, 2014 (accessed September 20, 2014). *http://www.virtuix.com*.

[363] D. Vogt, S. Grehl, E. Berger, H. Ben Amor, and B. Jung. A data-driven method for real-time character animation in human-agent interaction. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Intelligent Virtual Agents*, volume 8637 of *Lecture Notes in Computer Science*, pages 463–476. Springer International Publishing, 2014.

[364] C. Ware and R. Balakrishnan. Reaching for objects in vr displays: Lag and frame rate. *ACM Trans. Comput.-Hum. Interact.*, 1(4):331–356, December 1994.

[365] C. Ware and P. Mitchell. Reevaluating stereo and motion cues for visualizing graphs in three dimensions. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization*, APGV '05, pages 51–58. ACM, 2005.

[366] S. Weisband and S. Kiesler. Self disclosure on computer forms: meta-analysis and implications. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 3–10, New York, NY, USA, 1996. ACM.

[367] P. L. Weiss and N. Katz. The potential of virtual reality for rehabilitation. *Journal of Rehabilitation Research and Development*, 41(5):vii–x, Sep 2004.

[368] E. M. Wenzel, S. S. Fisher, P. K. Stone, and S. H. Foster. A system for three-dimensional acoustic "visualization" in a virtual environment workstation. In *Proceedings of the 1st Conference on Visualization '90*, VIS '90, pages 329–337, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[369] G. Wetzstein, M. Göllner, S. Beck, F. Weiszig, S. Derkau, J. P. Springer, and B. Fröhlich. Hector - scripting-based vr system design. In *ACM SIGGRAPH 2007 Posters*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[370] D.J. Wiley and J.K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Comput. Graph. Appl.*, 17(6):39–45, November 1997.

[371] P. Willemsen, M.B. Colton, S.H. Creem-Regehr, and W.B. Thompson. The effects of head-mounted display mechanical properties and field of view on distance judgments in virtual environments. *ACM Transactions on Applied Perception*, 6(2):8:1–8:14, March 2009.

[372] B. G. Wilson and K. M. Myers. Situated cognition in theoretical and practical context. *In Jonassen, D. H., & Land S. M. (Eds.) Theoretical foundations of learning environments*, pages 57–88, 2000.

[373] B.H. Wilson, R.R. Mourant, M. Li, and W. Xu. A virtual environment for training overhead crane operators: real-time implementation. *IIE Transactions*, 30(7):589–595, 1998.

[374] B.G. Witmer and P.B. Kline. Judging perceived and traversed distance in virtual environments. *Presence: Teleoperators and Virtual Environments*, 7(2):144–167, April 1998.

[375] D.F. Witmer, R.W. Colman, and S.L. Katzman. Measuring internet audiences: Patrons of an on-line art museum. *Doing internet research: Critical issues and methods for examining the net*, pages 163–178, 1999.

[376] R. Wojciechowski, K. Walczak, M. White, and W. Cellary. Building virtual and augmented reality museum exhibitions. In *Proceedings of the ninth international conference on 3D Web technology*, pages 135–144. ACM, 2004.

[377] D. Wollersheim, M. Merkes, N. Shields, P. Liamputtong, L. Wallis, F. Reynolds, and L. Koh. Physical and psychosocial effects of wii video game use among older women. *International Journal of Emerging Technologies and Society*, 8(2):85–98, 2010.

[378] Y.K. Wong, E. Hui, and J. Woo. A community-based exercise programme for older persons with knee pain using telemedicine. *Journal of telemedicine and telecare*, 11(6):310–315, 2005.

[379] Worldviz. Vizard vr software toolkit, 2014 (accessed September 20, 2014). *http://www.worldviz.com/products/vizard*.

[380] M. Wrzesien and M. Alcañiz Raya. Learning in serious virtual worlds: Evaluation of learning effectiveness and appeal to students in the e-junior project. *Computers and Education*, 55(1):178–187, August 2010.

[381] Y. Wu, S. V. Babu, R. Armstrong, J. W. Bertrand, J. Luo, T. Roy, S. B. Daily, L. C. Dukes, L. F. Hodges, and T. Fasolino. Effects of virtual human animation on emotion contagion in simulated inter-personal experiences. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):626–635, 2014.

[382] XSens. Xsens MVN, 2014 (accessed September 20, 2014). *http://www.xsens.com*.

[383] K.I. Yamaguchi, J.A. Takanishi. Development of a biped walking robot compensating for three-axis moment by trunk motion. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 561–566, 1993.

[384] K. Yamane, Y. Ariki, and J. Hodgins. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 169–178, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[385] D. Yang, J. Yu, and Y. Chung. Efficient compositing methods for the sort-last-sparse parallel volume rendering system on distributed memory multicomputers. *The Journal of Supercomputing*, 18(2):201–220, 2001.

[386] J. Yang, T. Sinokrot, K. Abdel-Malek, S. Beck, and K. Nebel. Workspace zone differentiation and visualization for virtual humans. *Ergonomics*, 51(3):395–413, 2008.

[387] Z. Yang, B. Yu, W. Wu, R. Diankov, and R. Bajscy. Collaborative dancing in tele-immersive environment. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 723–726, New York, NY, USA, 2006. ACM.

[388] S.H. You, S.H. Jang, Y. Kim, M. Hallett, S.H. Ahn, Y. Kwon, J.H. Kim, and M.Y. Lee. Virtual reality–induced cortical reorganization and associated locomotor recovery in chronic stroke an experimenter-blind randomized study. *Stroke*, 36(6):1166–1171, 2005.

[389] F. Zacharias, I.S. Howard, T. Hulin, and G. Hirzinger. Workspace comparisons of setup configurations for human-robot interaction. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3117–3122. IEEE, 2010.

[390] L. Zhao, A. Normoyle, S. Khanna, and A. Safonova. Automatic construction of a minimum size motion graph. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 27–35, New York, NY, USA, 2009. ACM.

[391] L. Zhao and A. Safonova. Achieving good connectivity in motion graphs. *Graph. Models*, 71(4):139–152, July 2009.

[392] Y. Zheng, Y. Hicks, D. Marshall, and D. Cosker. Real-time generation of interactive virtual human behaviours. In AlpeshKumar Ranchordas, HlderJ. Arajo, JooMadeiras Pereira, and Jos Braz, editors, *Computer Vision and Computer Graphics. Theory and Applications*, volume 24 of *Communications in Computer and Information Science*, pages 70–82. Springer Berlin Heidelberg, 2009.

[393] H. Zhou and H. Hu. Human motion tracking for rehabilitation a survey. *Biomedical Signal Processing and Control*, 3(1):1 – 18, 2008.