# InReach: Navigating and Manipulating 3D Models using Natural Body Gestures in a Remote Collaboration Setup

by
Anette Lia Freiin von Kapri

Diplom-Informatikerin
Rheinisch-Westfälische Technische Hochschule Aachen, 2008

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements of the degree of
Master of Science
at the Massachusetts Institute of Technology

June 2013

Signature of Author ____________________
Program in Media Arts and Sciences
May 10, 2013

Certified by ____________________
Pattie Maes
Associate Academic Head
Program in Media Arts and Sciences, MIT

Accepted by ____________________
Pattie Maes
Associate Academic Head
Program in Media Arts and Sciences, MIT

# InReach: Navigating and Manipulating 3D Models using Natural Body Gestures in a Remote Collaboration Setup

by
Anette Lia Freiin von Kapri

Diplom-Informatikerin
Rheinisch-Westfälische Technische Hochschule Aachen, 2008

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements of the degree of
Master of Science
at the Massachusetts Institute of Technology

June 2013

## ABSTRACT

Remote collaboration systems present audio and video representations of separate meeting spaces, but they do not support pointing towards and manipulating content in a shared digital space. InReach explores how remote collaborators can "reach into" a shared digital workspace where they can manipulate virtual objects and data. The collaborators see their live three-dimensional (3D) recreated mesh in a shared virtual space and can point at data or 3D models. They can grab digital objects with their bare hands, translate, scale, and rotate them. We discuss the design and implementation of the InReach system as well as application scenarios such as interior design, visiting virtual cities and studying 3D structures remotely. We report on results from a user study, which compares face-to-face and side-by-side arrangements for a sorting task. The user study demonstrates that different arrangements of collaborators and different level of detail for self-representation do not influence the feeling of co-presence significantly.

Thesis supervisor: Prof. Pattie Maes
Title: Associate Academic Head

# InReach: Navigating and Manipulating 3D Models using Natural Body Gestures in a Remote Collaboration Setup

by
Anette Lia Freiin von Kapri

Advisor ________________________________________

Pattie Maes
Associate Academic Head
Program in Media Arts and Sciences, MIT

Reader ________________________________________

V. Michael Bove, Jr.
Principal Research Scientist
Media Lab, MIT

Reader ________________________________________

Kent Larson
Principal Research Scientist
Media Lab, MIT

# ACKNOWLEDGMENTS

This work was possible through and improved by the support of a number of people to which I want to express my gratitude.

First and foremost my research advisor, Professor Pattie Maes. She invited me into her Fluid Interfaces group to work on natural ways to extend our body and mind. Through our brainstorming, discussions and suggestions I could sharpen my contributions which are reflected in this thesis work. I thank Michael Bove and Kent Larson for being readers for this thesis. Michael Bove gave me advice on technical aspects of my thesis which helped me in implementing the prototype. Kent Larson and some of his students showed me aspects in which an architect would be interested in, which I incorporated into my prototype. I would like to thank Seth Hunter, my collaborator. We started working together on the implementation with some sleepless nights and always had fruitful discussions regarding our overall visions. My UROP, Kevin Wong, for his hard work and long hours.

Thanks to Linda Peterson and Bill Tilden who kept me on schedule and reminded me of the various thesis deadlines. I would like to acknowledge Amna Carreiro to whom I could come with any non-research related questions.

I would like to express my gratidude towards Professor Steven Feiner who is a great academic advisor. He read early drafts of my thesis work and helped me figure out where to concentrate my efforts. Professor Hiroshi Ishii also gave me feedback on early work and encouraged me to think independently.

All my collaborators during the past two years have broadened my perspective. My fellow students, who are now also finishing their Master's Thesis. All the members of the Media Lab and especially the Fluid Interfaces Group keep challenging one another to produce the best results.

Finally, I would like to thank my family, my father and mother, who are always there for me, my two brothers Emanuel and Georg, the best and funniest brothers I could wish for. Tobias my partner in the mind and heart. And all my incredible friends, who keep me imaginative and humorous.

# TABLE OF CONTENTS

TABLE OF FIGURES ........ 11

1. INTRODUCTION ........ 16

1.1 MOTIVATION ........ 18

1.2 APPROACH ........ 19

1.3 CONTRIBUTIONS ........ 20

1.4 ROADMAP ........ 20

2. BACKGROUND AND RELATED WORK ........ 22

2.1 REMOTE COLLABORATION SYSTEMS THAT FOCUS ON THE INTERPERSONAL SPACE 23

2.2 REMOTE COLLABORATION SYSTEMS THAT FOCUS ON THE SHARED WORKSPACE ... 26

2.3 REMOTE COLLABORATION SYSTEMS THAT INTEGRATE INTERPERSONAL SPACE AND SHARED WORKSPACE ........ 27

2.4 3D MANIPULATION/HAND TRACKING ........ 33

3. USER INTERACTION DESIGN ........ 37

3.1 SELF-REPRESENTATION ........ 38

3.2 MANIPULATION ........ 38

3.3 NAVIGATION ........ 40

4. APPLICATIONS ........ 42

4.1 INTERIOR DESIGN ........ 42

4.2 VISITING CITIES VIRTUALLY WITH YOUR FRIENDS ........ 44

4.3 STUDYING 3D STRUCTURES ........ 46

5. IMPLEMENTATION ........ 47

5.1 SENSOR AND TRACKING ........ 48

5.2 HAND POSE DETECTION ........ 51

5.3 PHYSICS ........ 53

5.4 RENDERING ........ 54

5.5 NETWORK PROTOCOL ........ 55

6. USER STUDY AND EVALUATION ........ 56

6.1 PARTICIPANTS ........ 57

6.2 CONDITIONS ........ 57

*Face-to-Face (F2F)* ........ *57*
*Desktop VR (DVR)* ........ *58*
*Side-by-Side (SbS)* ........ *59*
6.3 SETUP/EQUIPMENT ........ 59
6.4 EXPERIMENT PROCEDURES ........ 60
6.5 MEASURES ........ 61
6.6 EVALUATION ........ 62
**7. CONCLUSIONS AND FUTURE WORK ........ 67**
**BIBLIOGRAPHY ........ 69**
**ACRONYMS ........ 73**
**APPENDIX A: QUESTIONNAIRE ........ 74**
**APPENDIX B: CODE SAMPLES ........ 79**
B.1 HAND POSE DETECTION ........ 79
B.2 SETTING UP THE ADAPTIVE NAÏVE BAYES CLASSIFIER ........ 82

## Table of Figures

Figure 1. InReach is a remote collaboration system that projects multiple collaborators into the same 3D space. The picture shows the local person (on the left) in a mirrored view together with the remote person in the same environment. .............. 18

Figure 2. CSCW systems are classified along two axes: place and time. The collaboration can happen in the same or different place as well as same or different time. Image courtesy of Johansen (Johansen, 1988)................................................................. 22

Figure 3. Cisco's telepresence room extends a round table into the remote spaces to have participants sit at the same table. It uses high-definition video and audio. Image courtesy of Cisco..................................................................................... 24

Figure 4. This image shows Polycom's telepresence system. It allows remote collaborators to sit together at a round table. Image courtesy of Polycom.............. 24

Figure 5. HyperMirror (Morikawa & Maesako, 1998) is a system that projects both of the distant collaborators onto the same screen using a green screen setup. Image courtesy of Morikawa and Maesako. ................................................................... 25

Figure 6. Reflection of presence (Agamanolis & Bove, Jr., 1997) is a telepresence system in which collaborators can see their own reflection as well as reflections from other participants on the screen. Based on speech and movement of the participants, they are brought to the foreground or drawn transparently. Image courtesy of Agamanolis and Bove, Jr. ................................................................................................ 25

Figure 7. In the side-by-side telepresence system (Tanner & Shah, 2010) two remote workstations are prototyped to simulate a side-by-side work environment. The illusion is created that users are sitting next to each other. Image courtesy of Tanner and Shah.................................................................................................... 26

Figure 8. Oblong's Mezzanine is a multi-screen conference room. Participants can share their private screen publicly and move slides into the presentation deck. Remote users are visible as a video stream. Image courtesy of Oblong. .............................. 27

Figure 9. The ClearBoard system (Ishii, Kobayashi, & Arita, 1994) allows two remote users to collaboratively draw on a shared virtual surface. Image courtesy of Ishii, Kobayashi and Arita. ........................................................................................ 28

Figure 10. VideoArms (Tang, Neustaedter, & Greenberg, 2007) digitally recreates people's body actions as virtual embodiments in a remote workspace. Here, two groups of two people work remotely on two connected displays. Image courtesy Tang, Neustaedter and Greenberg ........ 28

Figure 11. Colla-Board (Nescher & Kunz, 2011) overlays the remote life-sized video image atop the shared common whiteboard. It is keeping the whiteboard's content editable at both sides. Image courtesy of Nescher and Kunz. ........ 29

Figure 12. A conceptual sketch and the implementation of the office of the future (Raskar, Welch, Cutts, Lake, Stesin, & Fuchs , 1998). Image courtesy of State, Raskar, Welch, Cutts, Lake, Stesin and Fuchs. ........ 30

Figure 13. KeckCAVES is a science visualization tool with remote collaboration capabilities. Remote users are projected into a virtual space and can inspect a dataset together. Image courtesy of Kreylos. ........ 30

Figure 14. In MirageTable (Benko, Jota, & Wilson, 2012), a 3D stereoscopic projector projects content directly on top of the curved screen. Remote collaborators are sitting in a face-to-face configuration. Image courtesy of Benko, Jota, and Wilson.31

Figure 15. ARCADE (Stein, Xiao, Tompson, Hendee, Perlin, & Ishii, 2012) is a system that allows real-time video-based presentations that convey the illusion that presenters are directly manipulating holographic 3D objects with their hands. Image courtesy of Stein, Xiao, Tompson, Hendee, Perlin and Ishii. ........ 32

Figure 16. Maimone et al. (Maimone, Bidwell, Peng, & Fuchs, 2012) developed a telepresence system which uses multiple Kinect cameras and recreates a 3D scene. They optimize the mesh generation and mesh merging using GPU-accelerated data processing. Image courtesy of Maimone, Bidwell, Peng, and Fuchs. ........ 33

Figure 17. zSpace combines a stereoscopic 3D screen with a tracked pen-like devide. This allows a user to manipulate virtual 3D objects precisely. Image courtesy of zSpace. ........ 34

Figure 18. The Leonar3Do Bird is an accessory for a computer, that enables head and device tracking. A user can then explore 3D virtual data with the device. Image courtesy of Leonar3Do. ........ 34

Figure 19. 6D hands is a bimanual hand tracking system that provides physically-motivated 6-DOF control for 3D assembly. Image courtesy of Wang, Paris and Popovic. .......... 35

Figure 20. The hand tracking algorithm presented in (Oikonomidis, Kyriazis, & Argyros, 2011) uses a single Kinect camera to track 27 DOF of the hand. It recreates a virtual representation of the hand but needs a higher-end GPU for real-time performance. Image courtesy of Oikonomidis, Kyriazis and Argyros. .......... 36

Figure 21. InReach consists of two remote setups. One user is in one location the other is located somewhere else. Each user can move freely in front of the Kinect-screen-setup in his/her physical space. On screen in the digital environment both users, the remote and the local one, are visible. They are standing next to each other. .......... 37

Figure 22. (a) Open hand for no action. (b) Single hand grab to select object. Moving the hand will then translate the object. (c) Bimanual interaction for scaling and rotation. Scaling based on distance between hands. (d) Rotation based on axis defined by hands. .......... 39

Figure 23. PenguFly (von Kapri, Rick, & Feiner, 2011) is a body-directed travel technique. The standing user's head and hands are tracked. Their positions are projected onto the ground to define a 2D triangle. Travel direction is defined by d, whereas the velocity depends on the length of d. .......... 40

Figure 24. (left) A virtual model of a living room. (right) Two remote users are projected into that living room. One person is pretending to sit on the couch. .......... 42

Figure 25. This shows the setup of the system. The local user can be seen from the back and on the screen both users are visible standing in the virtual living room. .......... 43

Figure 26. In the interior design scenario a user can move around a room and place furniture somewhere else as well as rotate it around. Here you see a user grabbing a dining table and moving it to a different location in the room as well as turning it. 44

Figure 27. A user is flying around a simplified virtual model of the city of Munich by leaning into the direction s/he wants to fly to. .......... 45

Figure 28. Two users are exploring the virtual city together. The front user takes control of the navigation. .......... 45

Figure 29. A conceptual application focuses on an educational game in which a teacher and her/his students can "inhabit" a protein and point out particular components and features. ........ 46

Figure 30. This diagram shows the architecture of the InReach framework. Two stations are connected over the network; a UDP connection is set up to send the RGB and depth images, and the OSC connection transfers meta data such as user and skeleton information, hand poses and hand actions as well as states of virtual objects over the network. ........ 47

Figure 31. The Microsoft Kinect sensor captures a color as well as a depth image in the resolution of 640x480. Image courtesy of Microsoft ........ 48

Figure 32. The Kinect captures the color image (left) every frame as well as the depth image (right) at a rate of 30 FPS. ........ 49

Figure 33. (left) Different users are distinguished based on the depth image. (right) The 3D meshes of the users are generated from the color and depth image. If the user goes into an initial tracking pose his/her skeleton will be tracked consequently. .... 50

Figure 34. To extract the user's hand mask only depth values around an area of the skeleton's (x, y, z)-hand-position are considered. ........ 51

Figure 35. (1) Open hand. (2) Closed hand. (3) Open hand sideways. (a) Hand mask. (b) Hand contour. (c) Convexity defect. (d) Minimal fitting ellipse. ........ 52

Figure 36. A user is grabbing a cube and pushing away two other ones. This sequence shows the effect of the physics. ........ 54

Figure 37. (left) Using parallel axis asymmetric frusta for the stereoscopic rendering aligns the projection planes for the left and right eye on top of each other. (right) Regular perspective projection frusta would not overlay the projections planes of the left and right eye in a stereoscopic setup. ........ 54

Figure 38. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Face-to-Face (F2F) setup the local user can only see his/her hands as circles in the environment. (left) Actual view. (right) F2F setup. ........ 58

Figure 39. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Desktop VR (DVR) setup the local user can see his/her skeleton from the back. The arms are red and the rest of the body is gray. (left) Actual view. (right) DVR setup. 58

Figure 40. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Side-by-Side (SbS) setup the local user sees a mirrored 3D mesh of him/herself. (left) Actual view. (right) SbS setup. 59

Figure 41. Subjective rating of frustration and intuitiveness for the 3D manipulation experience over all conditions (error bars show standard deviation). 63

Figure 42. (left) Subjective preferences of conditions in terms of co-presence ranking. The three conditions are Face-to-Face (F2F), Desktop VR (DVR), and Side-by-Side (SbS). (right) Subjective preferences of conditions in terms of game play rating. (Error bars show standard deviation.) 64

# 1. INTRODUCTION

In times of globalization, colleagues and collaborators are often distributed all over the world. An ordinary workday often involves meetings at which at least one person is in a remote location. With the increasing availability and decreasing cost of Internet bandwidth, video cameras, computers, and microphones, the barriers to designing more effective remote collaboration systems have been lowered. However, several problems persist in state of the art systems used for remote collaboration. These include visually disjunctive physical and virtual spaces, limited ways to employ gestures as a communication tool, and limited ways to collaboratively mark and manipulate shared content.

In this thesis we will look at a subset of remote collaboration meetings and want to distinguish two types of those. These two differ in their focus. A first type of meeting focuses on critical decision-making, while a second type of meeting focuses on the collaborative creating and modifying of digital content. Examples of the latter type of meeting include brainstorming, creating a joint presentation, creating designs and models, etc. In a decision-making meeting, collaborators might prefer to have the sense of being in the same room and talking to the remote person as if they were talking face-to-face thereby concentrating on the interpersonal space. In such a context, the attention lies on displaying a realistic representation of the remote person on the screen, correct as to their physical size and gaze direction, and extending the remote physical space into the local place. High quality audio may be just as important. In contrast, the second type of meeting includes creative meetings that may be more directed towards brainstorming or discussing a computer-aided design (CAD) model. They instead focus on the data itself, the shared workspace. For these meetings, it is less important to imitate an actual meeting room setting and provide high fidelity sound and image of the remote participants. Instead, the emphasis is on the object of creation or the data that is being inspected and worked on.

Hollan and Stornetta (Hollan & Stornetta, 1992) discuss inherent problems in the approach of designing remote collaboration systems that imitate face-to-face communication: " [The] imitation will never be as good as the real thing." They propose to focus on the "communication" aspect instead of the "tele" aspect, so that people prefer to use the system, physically proximate or not.

We are interested in the representation and integration of the interpersonal space and the shared workspace in one and the same system. This thesis introduces InReach (Figure 1), a system that explores how remote collaborators can "reach into" a shared digital space where they can see one another, as well as manipulate virtual objects and data. The collaborators see their 3D representations in a shared 3D virtual space. Their 3D meshes are displayed next to each other, which creates the illusion of an augmented mirror. In this mirror you can see yourself as well as your collaborator inside the models and information which you are collaborating on. It breaks down the virtual curtain that separates collaborators in a face-to-face setting. Collaborators can point at and manipulate the shared models and data using natural hand gestures.

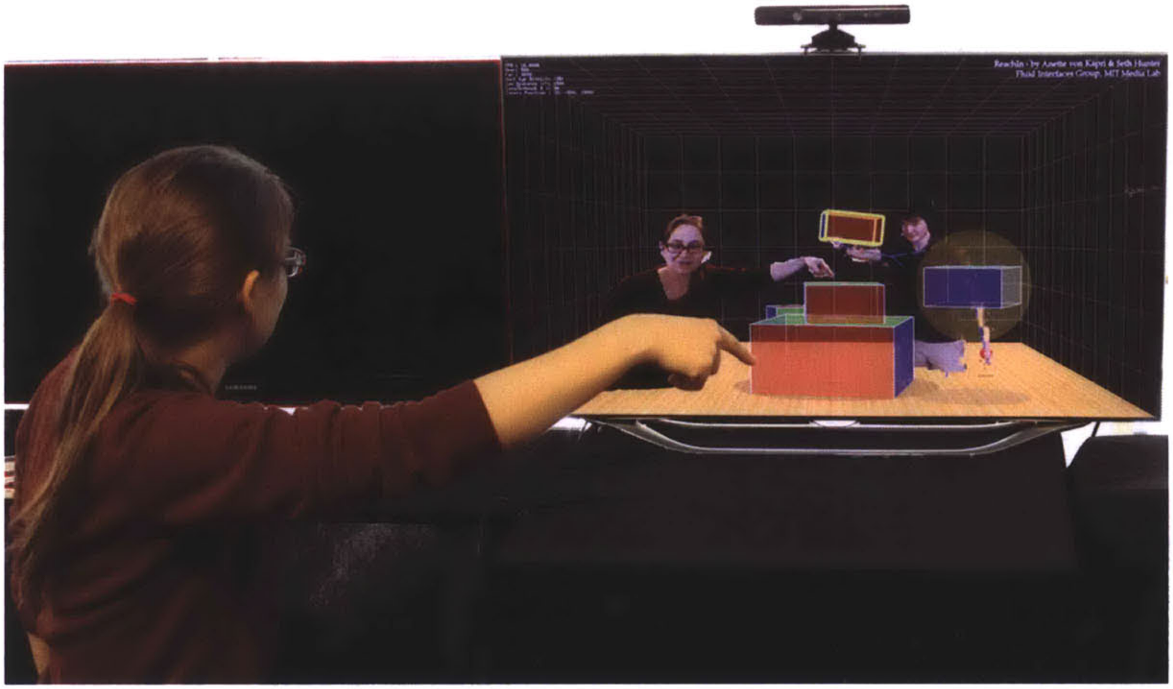

ReachIn - by Anette von Kapri & Seth Hunter
Fluid Interfaces Group, MIT Media Lab

**Figure 1. InReach is a remote collaboration system that projects multiple collaborators into the same 3D space. The picture shows the local person (on the left) in a mirrored view together with the remote person in the same environment.**

### *1.1 Motivation*

InReach is motivated by the idea of users not having to split their attention between two windows, one containing shared data and the other containing a live image of the remote collaborator. Instead there is one "window" where the user sees him/herself, the collaborator, and the information on which they want to work collaboratively. Possible advantages are that users do not have to split their attention, that they can use their body to point to and manipulate the data with their bare hands and natural gestures, and that they can see themselves in relation to the data. This is especially useful for applications such as architectural, interior, and landscape design; applications where the collaboration revolves around the body, such as theater and dance; and joint presentations that would benefit from a shared virtual space for two or more remote collaborators.

### *1.2 Approach*

Paul Dourish gives a good description of phenomenologists' views and how they relate to human computer interaction (HCI) in his book "Where The Action Is: The Foundations of Embodied Interaction" (Dourish, 2001). The mind and the body define the human experience. Being-in-the-World means that we encounter the world directly rather than abstractly, but this direct experience is not reflected in how technology and interfaces have evolved.

How can we integrate the body back into our interactions with technology? Can we move away from the mind-focused, more theoretical perception we find in computers today to integrate bodily interaction? By integrating our physical body into the digital world, we can become more engaged in our interactions with technology but also be more effective and efficient at what we do.

The InReach project concentrates on exploring this idea in the area of remote collaboration. The goal is to enhance the feeling of co-presence by augmenting the feeling of being-there through selfmovement. Users are encouraged to move around a shared virtual space by moving physically in their space. By reaching out to a virtual object they can interact with it. This higher degree of bodily engagement might increase the feeling of co-presence.

InReach represents one approach for including proprioception and embodied interaction into an environment for remote collaboration. It explores how remote collaborators can "reach into" a shared digital workspace where they can manipulate virtual objects and data. The collaborators see their live 3D recreated mesh in a shared virtual space on a large screen in front of them and can use their bodies to “inhabit” and interact with 3D models. They can navigate and move their image inside of the 3D model. They can also grab digital objects with their bare hands, translate, scale, and rotate them. In contrast with the traditional view for remote conferencing InReach is particularly useful for situations in which users can benefit from seeing their own and their collaborator’s body in relation to the data and can use their bodies to navigate and manipulate the data.

### *1.3 Contributions*

Our contributions are trifold: (1) the design and implementation of a novel system for remote collaboration, (2) the application scenarios as well as a prototype 3D remote collaboration application, and (3) the results from an exploratory user study that examines differences in self-representation of the local users. Our experiments showed that different level-of-details for self-representation and different arrangements of collaborators do not show significant differences in the feeling of co-presence.

### *1.4 Roadmap*

After giving an overview of related work regarding collaboration systems focusing on either the interpersonal space or the shared workspace or a combination and 3D manipulation techniques in Chapter 2. BACKGROUND AND RELATED WORK we first describe the experience of using the InReach system in Chapter 3. USER INTERACTION DESIGN. This relates to the self-representation, the manipulation of 3D objects as well as the navigation in the virtual space. Then in Chapter 4. APPLICATIONS we give an overview of possible usage scenarios for such a remote collaboration setup such as interior design exploration, visiting virtual cities together, and explaining 3D processes and structures remotely. Chapter 5. IMPLEMENTATION introduces the overall system architecture with all its individual parts such as the sensor and tracking, the hand pose detection, the network protocol, and the physics. We present a user study which compares different modes of self-representation and measures the feeling of co-presence as well as intuitiveness and frustration of the interaction metaphor in Chapter 6. USER STUDY AND EVALUATION. As a final outlook we conclude the thesis with Chapter 7. CONCLUSIONS AND FUTURE WORK.

## 2. BACKGROUND AND RELATED WORK

Johansen (Johansen, 1988) classifies computer-supported cooperative work (CSCW) systems along two axes: place and time (see Figure 2). The collaboration can happen in the same or in a different place and it can happen at the same or at a different time. InReach focuses on collaborations that happen at the same time but in different places.

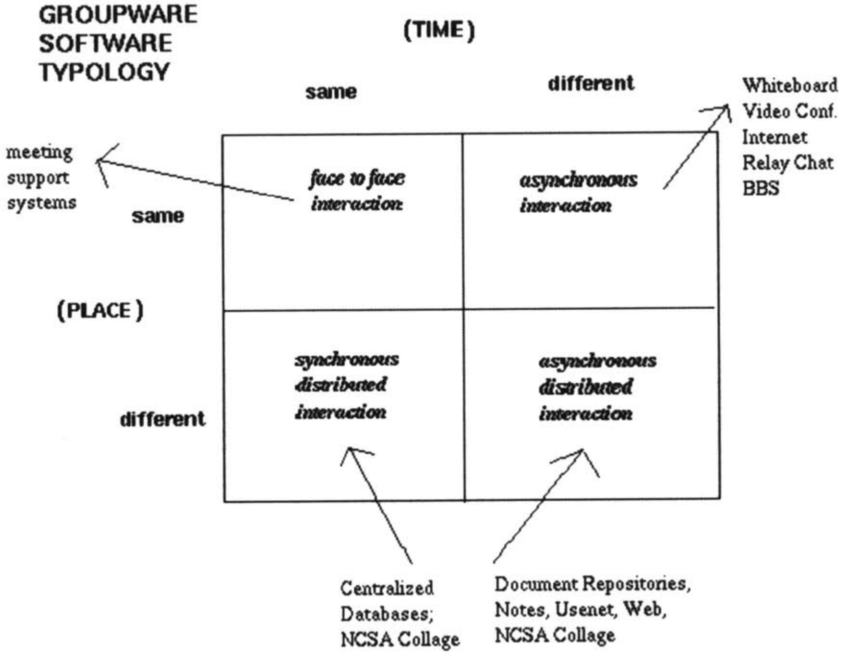


**Figure 2. CSCW systems are classified along two axes: place and time. The collaboration can happen in the same or different place as well as same or different time. Image courtesy of Johansen (Johansen, 1988).**

These synchronous distributed systems can further be categorized into audioconferencing, groupware, videoconferencing, telepresence and collaborative mixed reality systems (Wolff, Roberts, Steed, & Otto, 2007). The review of related work below classifies remote collaboration systems according to whether they prioritize the interpersonal space, the shared workspace, or an integration of both. The last subsection discusses related work in the area of 3D manipulation and gesture tracking.

### *2.1 Remote Collaboration Systems that Focus on the Interpersonal Space*

A first category of systems concentrate on the interpersonal space, they focus on how to enhance the experience of the personal connection and communication between the people participating in the remote meeting. Commercially available video conferencing systems typically enable a single user to broadcast visual information to a group. For example, using applications such as Skype (Skype), users can share their captured camera image with others. Remote users are connected through the video and audio streams of each other.When is comes to discussing data, users can share their screen with others. Broadcasting gives control to a single user at any one time. Furthermore, there is a separation between the video stream and other visual information (the topic of the conversation), which are often presented in different screen-based windows.

Even Cisco's (Cisco TelePresence) or Polycom's (Polycom TelePresence) telepresence systems (see Figure 3 and Figure 4), high-end video conferencing systems that attempt to recreate many aspects of in-person meetings, project data from a single user on a separate screen below the video feed of the remote participants. Such applications are more concentrated on creating the illusion of having remote participants sitting at the same table and attempt to recreate a real-world setting. But they limit gestural interaction by not allowing users to directly gesture to the data or annotate information collaboratively.

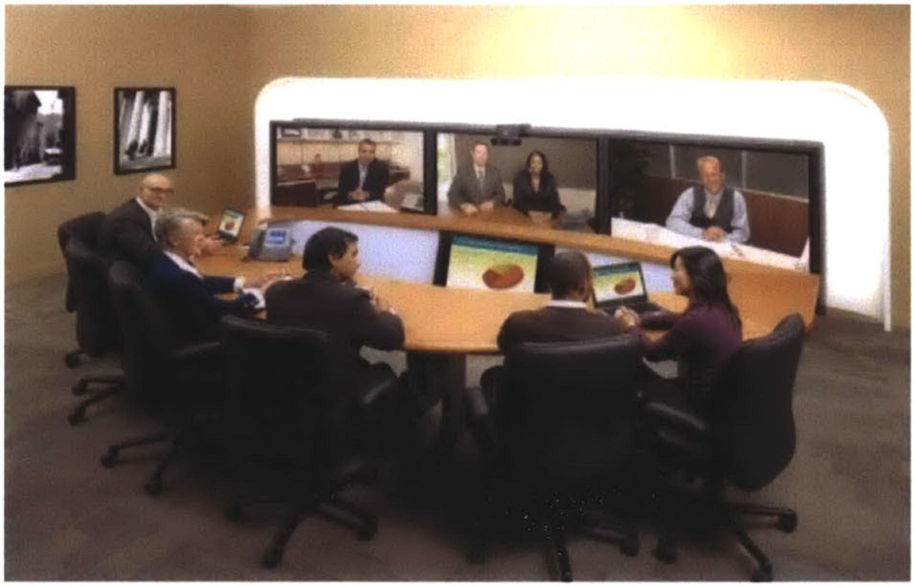


**Figure 3. Cisco's telepresence room extends a round table into the remote spaces to have participants sit at the same table. It uses high-definition video and audio. Image courtesy of Cisco.**

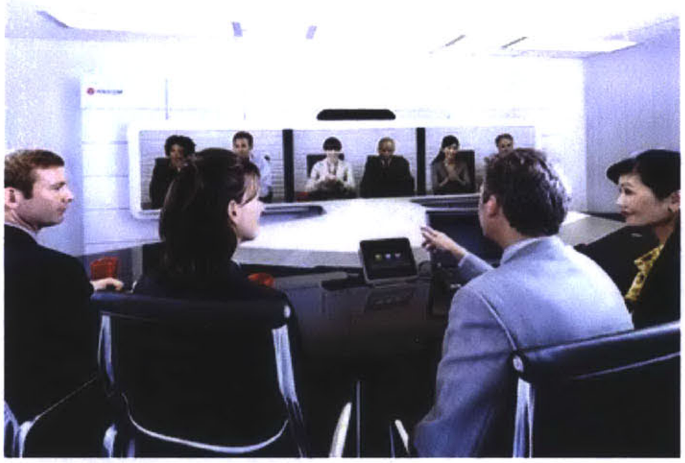


**Figure 4. This image shows Polycom's telepresence system. It allows remote collaborators to sit together at a round table. Image courtesy of Polycom.**

Morikawa et al. presented HyperMirror (Morikawa & Maesako, 1998) a system that projects both of the distant collaborators onto the same screen using a green screen setup (see Figure 5). HyperMirror does not try to recreate a face-to-face meeting but places both users into a shared space. It concentrates on the interpersonal space and does not add any digital content as our system does.

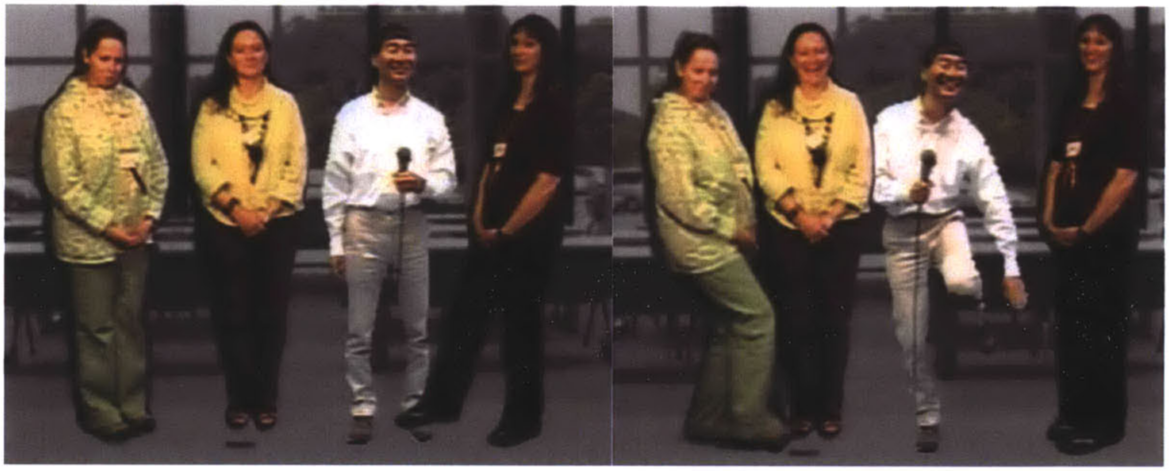


**Figure 5. HyperMirror (Morikawa & Maesako, 1998) is a system that projects both of the distant collaborators onto the same screen using a green screen setup. Image courtesy of Morikawa and Maesako.**

Reflection of presence (Agamanolis & Bove, Jr., 1997) places remote collaborators onto the same screen next to each other. Additionally, this system focuses on the participant who is in the center of attention. It analyzes speech and movement and varies the scale, position and transparency (see Figure 6).

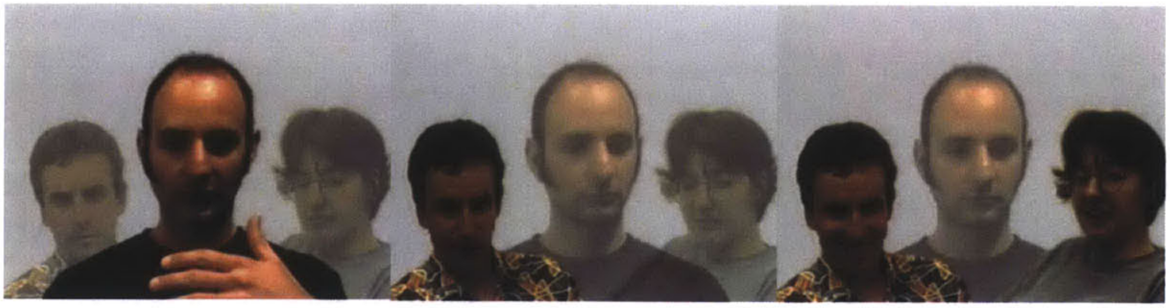


**Figure 6. Reflection of presence (Agamanolis & Bove, Jr., 1997) is a telepresence system in which collaborators can see their own reflection as well as reflections from other participants on the screen.**

**Based on speech and movement of the participants, they are brought to the foreground or drawn transparently. Image courtesy of Agamanolis and Bove, Jr.**

### *2.2 Remote Collaboration Systems that Focus on the Shared Workspace*

A second type of remote collaboration system enables more synchronous control by allowing multiple users to annotate or edit a document together. A popular system is Google Docs, in which the remote participant is just represented as a pointer. Another commercial example of such a system is Cisco's WebEx (Cisco webex) in which meeting participants can share and edit presentation slides. However, in this application, there is still a physical separation between the video feed and the data, preventing direct gesturing to the data. While in Google docs there is no visual representation of the participant other than the cursor.

Tanner and Shah's (Tanner & Shah, 2010) side-by-side telepresence system places the collaborator on a separate screen next to the user to create a feeling of sitting next to each other (see Figure 7). Collaborators communicate but focus mainly on their common task.

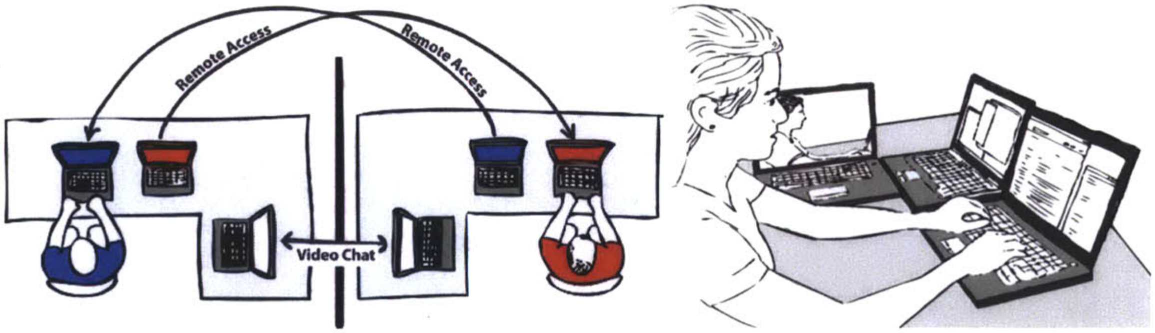


**Figure 7. In the side-by-side telepresence system (Tanner & Shah, 2010) two remote workstations are prototyped to simulate a side-by-side work environment. The illusion is created that users are sitting next to each other. Image courtesy of Tanner and Shah.**

Oblong's Mezzanine system (Oblong g-speak/Mezzanine) as depicted in Figure 8 concentrates on integrating multiple users into an interactive workflow sharing whiteboards and slides. A remote user's video is one data asset, but users are not directly integrated in the data and as such cannot combine the interpersonal space with the shared workspace.

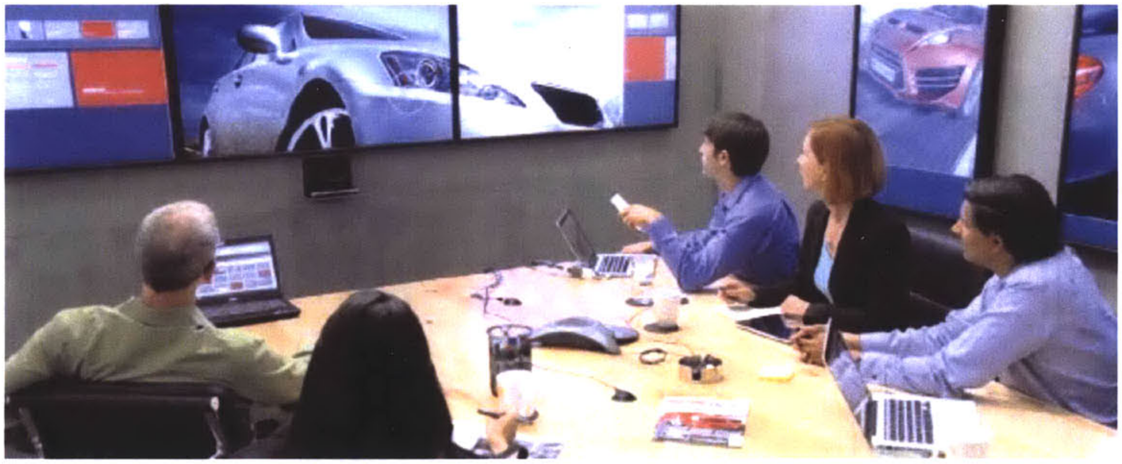


**Figure 8. Oblong's Mezzanine is a multi-screen conference room. Participants can share their private screen publicly and move slides into the presentation deck. Remote users are visible as a video stream. Image courtesy of Oblong.**

### *2.3 Remote Collaboration Systems that integrate Interpersonal Space and Shared Workspace*

Several research projects have overlaid live video and data to enable direct gesture and gaze interaction. An early example of such technology is ClearBoard (Ishii, Kobayashi, & Arita, 1994), a system which allows two remote users to collaboratively draw on a shared virtual surface (see Figure 9). The resulting system seamlessly integrates the interpersonal space and the shared workspace.

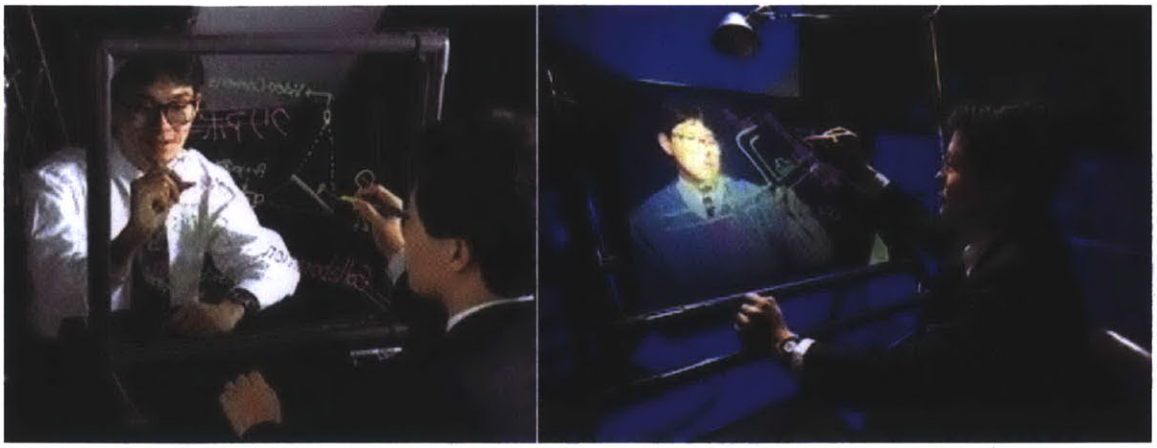


**Figure 9. The ClearBoard system (Ishii, Kobayashi, & Arita, 1994) allows two remote users to collaboratively draw on a shared virtual surface. Image courtesy of Ishii, Kobayashi and Arita.**

VideoArms (Tang, Neustaedter, & Greenberg, 2007) is an embodiment technique that captures and reproduces people's arms as they work over large displays which promotes awareness of remote collaborators (see Figure 10).

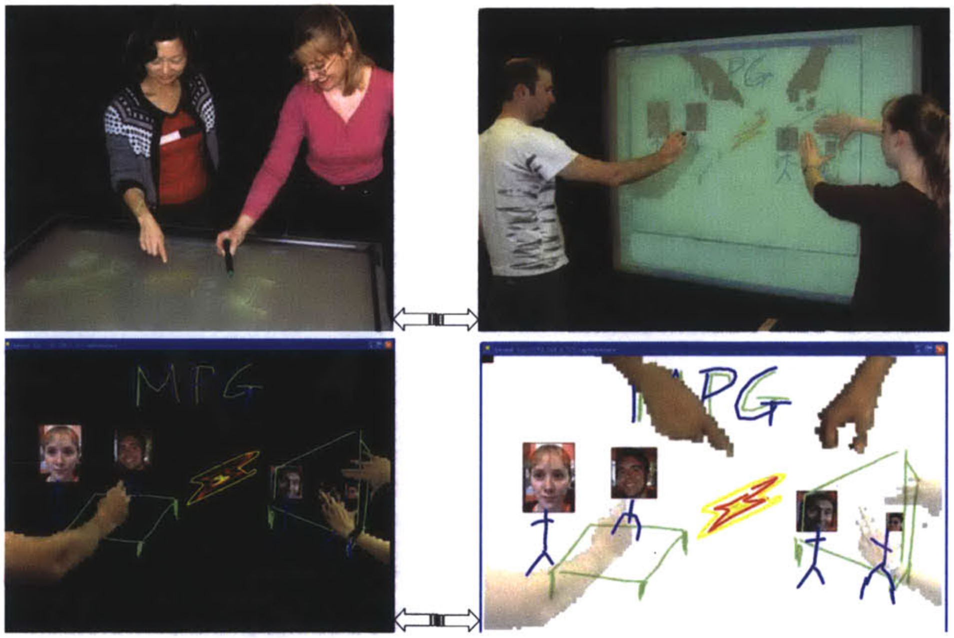


**Figure 10. VideoArms (Tang, Neustaedter, & Greenberg, 2007) digitally recreates people's body actions as virtual embodiments in a remote workspace. Here, two groups of two people work remotely on two connected displays. Image courtesy Tang, Neustaedter and Greenberg.**

Another example is Colla-Board (Nescher & Kunz, 2011) shown in Figure 11, a collaborative whiteboard system in which users are projected onto the whiteboard surface. The difference with the research discussed in this thesis is that for all of these examples users are only collaborating on 2D documents.

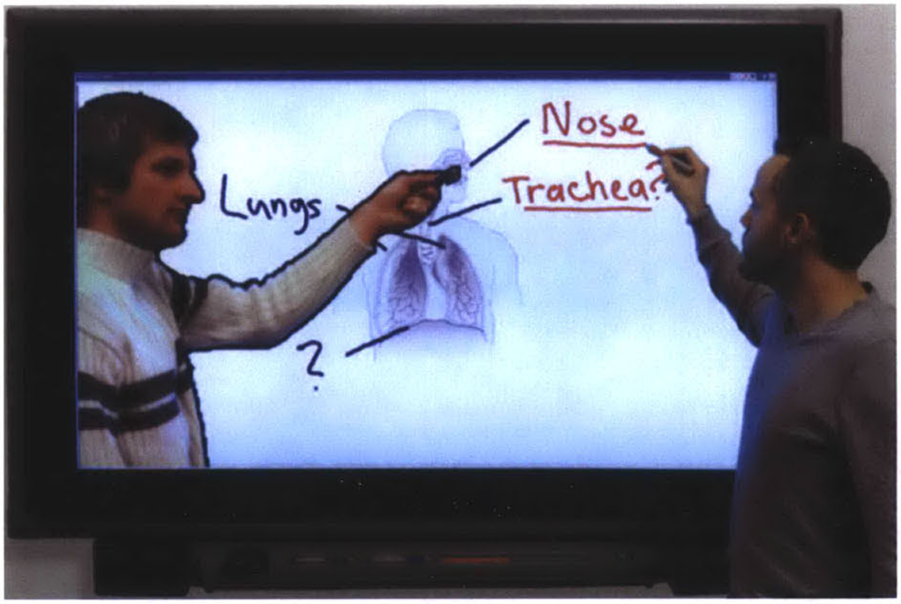


**Figure 11. Colla-Board (Nescher & Kunz, 2011) overlays the remote life-sized video image atop the shared common whiteboard. It is keeping the whiteboard's content editable at both sides. Image courtesy of Nescher and Kunz.**

The office of the future (Raskar, Welch, Cutts, Lake, Stesin, & Fuchs , 1998) concept describes how we could collaboratively manipulate virtual, 3D objects from our office desks extending the real office with projected images of a remote office and virtual objects. With the recent developments of commercially available low-cost depth sensors such as the Kinect (Microsoft Kinect), this vision becomes more real.

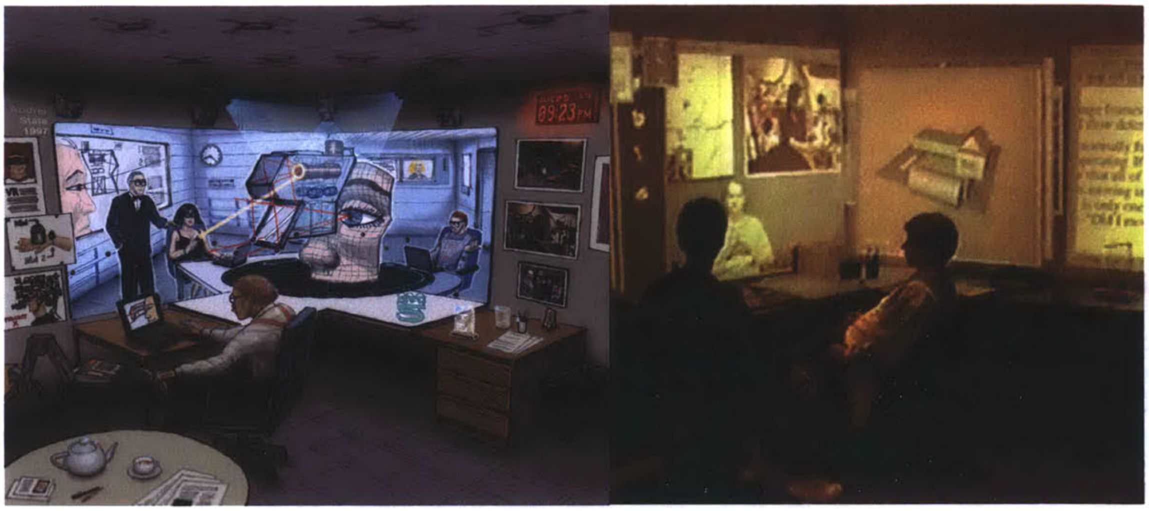


**Figure 12. A conceptual sketch and the implementation of the office of the future (Raskar, Welch, Cutts, Lake, Stesin, & Fuchs , 1998). Image courtesy of State, Raskar, Welch, Cutts, Lake, Stesin and Fuchs.**

KeckCAVES (W.M. Keck Center for Active Visualization in the Earth Sciences) is a collaborative science visualization tool for Virtual Reality (VR). 3D meshes of remote users can be projected into a virtual space together with the dataset (see Figure 13). Setups use high-end VR systems, which cannot be distributed to the masses.

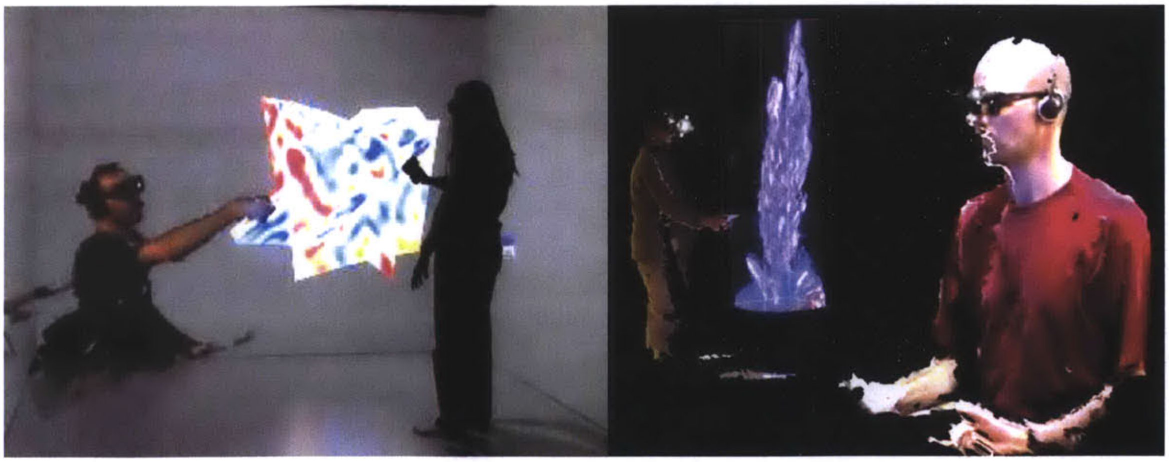


**Figure 13. KeckCAVES is a science visualization tool with remote collaboration capabilities. Remote users are projected into a virtual space and can inspect a dataset together. Image courtesy of Kreylos.**

MirageTable (Benko, Jota, & Wilson, 2012) mimics the situation of two collaborators working at a table (see Figure 14). The 3D mesh of the remote user is projected onto a desk that curves upward, giving the illusion that the users face each other. Users can create virtual copies of real objects. A digital representation of each user's hands is created, with which they can interact with virtual objects in a physical simulation.

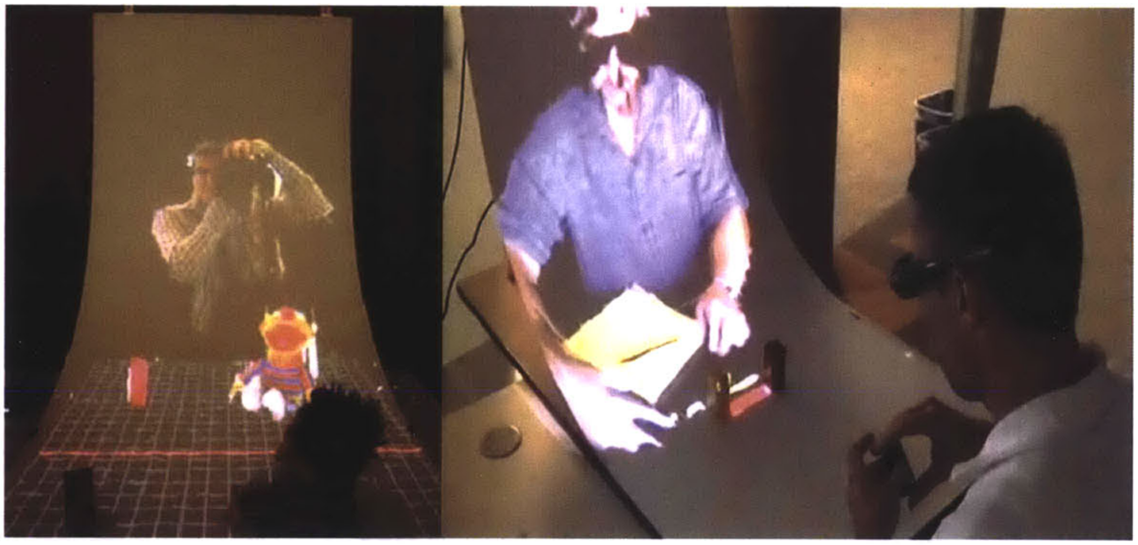


**Figure 14. In MirageTable (Benko, Jota, & Wilson, 2012), a 3D stereoscopic projector projects content directly on top of the curved screen. Remote collaborators are sitting in a face-to-face configuration. Image courtesy of Benko, Jota, and Wilson.**

ARCADE (Stein, Xiao, Tompson, Hendee, Perlin, & Ishii, 2012) allows remote video-based presentation (see Figure 15). Virtual 3D objects can be placed on top of the video of the remote collaborator who can directly manipulate them with his/her hands. The fingers are tracked in 3D using the Kinect. The difference with our system is that these approaches try to mimic reality as closely as possible. They do not explore other ways of self-representation than we know from our daily interactions.

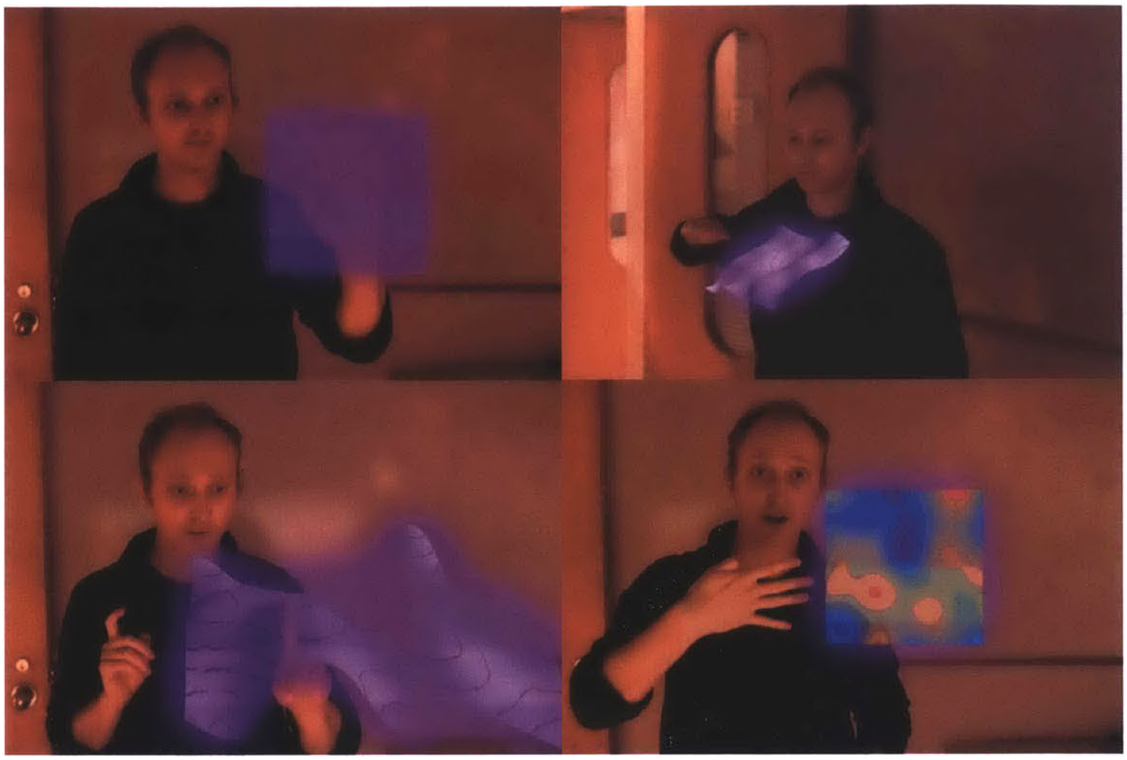


**Figure 15. ARCADE (Stein, Xiao, Tompson, Hendee, Perlin, & Ishii, 2012) is a system that allows real-time video-based presentations that convey the illusion that presenters are directly manipulating holographic 3D objects with their hands. Image courtesy of Stein, Xiao, Tompson, Hendee, Perlin and Ishii.**

Maimone et al. (Maimone, Bidwell, Peng, & Fuchs, 2012) present an autostereoscopic telepresence system that offers fully dynamic, real-time 3D scene capture and continuous-viewpoint, head-tracked stereo 3D display. They focus on the technical implementation and describe how to diminish the interference problems you get using multiple Kinect cameras, how to optimize the meshes through hole filling and smoothing and getting rid of discontinuous surfaces on the graphics processing unit (GPU) and merge meshes from different cameras. Their implementation does not allow users to directly interact with digital objects as our system does. See Figure 16.

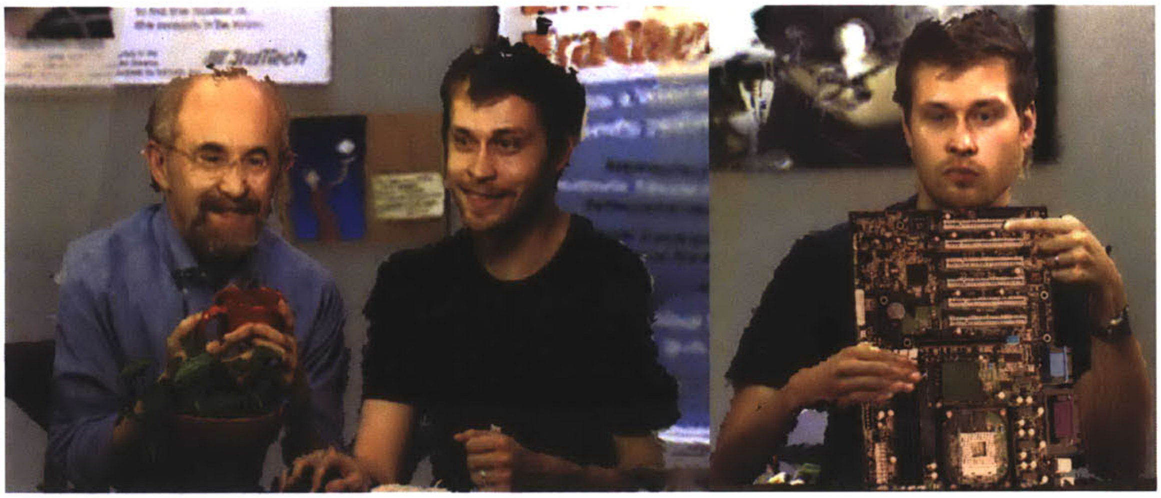


**Figure 16. Maimone et al. (Maimone, Bidwell, Peng, & Fuchs, 2012) developed a telepresence system which uses multiple Kinect cameras and recreates a 3D scene. They optimize the mesh generation and mesh merging using GPU-accelerated data processing. Image courtesy of Maimone, Bidwell, Peng, and Fuchs.**

### *2.4 3D Manipulation/Hand Tracking*

Hand gestures are an expressive form of human communication and interaction in the real world. We use our hands to point to objects in the environment, which gives people present a context for the discussion. We can grab and lift things and interact with the real world. In a remote setting where collaborators are talking about a shared dataset it would be useful to be able to track the hands in all their freedom to allow for an intuitive interaction with the datasets.

Commercial products such as zSpace (zSpace by Infinite Z) (see Figure 17) and the Leonar3Do bird (Bird by Leonar3Do) (see Figure 18) allow the user to grab virtual objects behind a stereoscopic 3D screen with a tracked pen-like device. This allows for high precision in manipulation tasks such as constructing an industrial model from individual pieces. Such devices serve as a bridge between the real and the digital world.

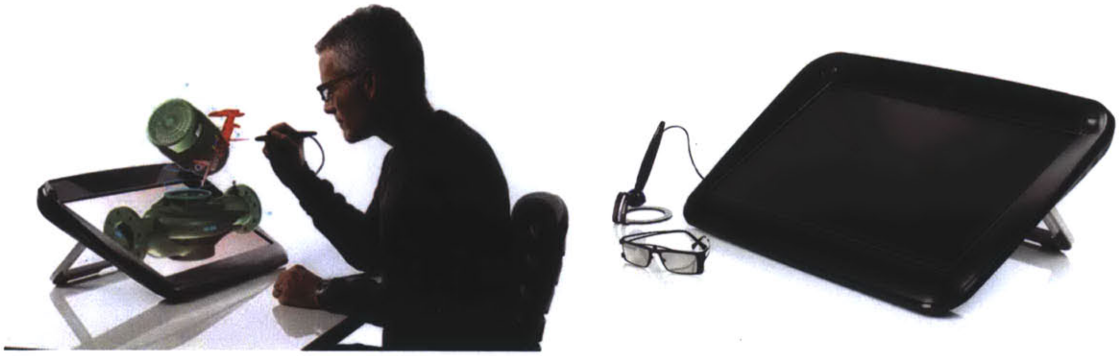


**Figure 17. zSpace combines a stereoscopic 3D screen with a tracked pen-like devide. This allows a user to manipulate virtual 3D objects precisely. Image courtesy of zSpace.**

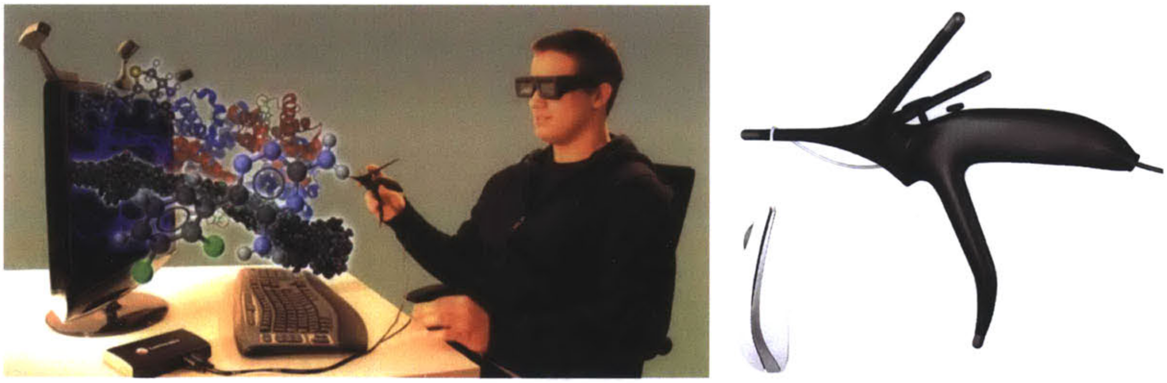


**Figure 18. The Leonar3Do Bird is an accessory for a computer, that enables head and device tracking. A user can then explore 3D virtual data with the device. Image courtesy of Leonar3Do.**

To allow for more intuitive interactions it would be great if we could simply use our body to "touch" the 3D digital world. Wang and Popovic (Wang & Popovic, 2009) presented a colored glove with which the hand's 3D pose and configuration can be tracked in real-time. The hand can be used as input in desktop VR applications, for example. Still, the user has to wear an additional device, the glove, to interact with the system. Wang et al. later proposed "6D hands" (Wang, Paris, & Popovic, 2011) (see Figure 19), a markerless hand-tracking system using two webcams. The system can track position and orientation of multiple hands as well as recognizing pinching gestures. It does not model the fingers explicitly but runs at interactive rates of 20Hz on an Intel Core i7.

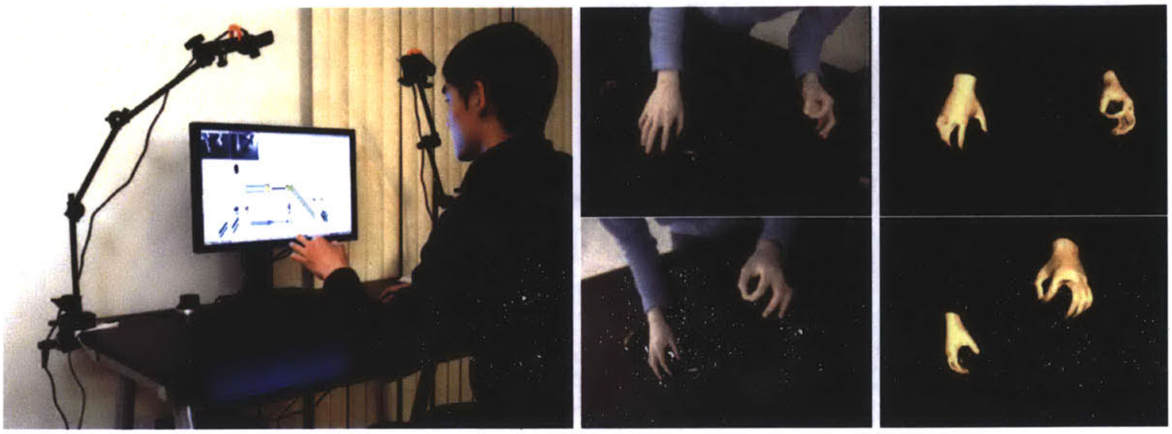


**Figure 19. 6D hands is a bimanual hand tracking system that provides physically-motivated 6-DOF control for 3D assembly. Image courtesy of Wang, Paris and Popovic.**

Oikonomidis et al. (Oikonomidis, Kyriazis, & Argyros, 2011) presented a hand-tracking algorithm using a single Kinect camera. It can track the 27 degrees of freedom (DOF) of the hand and creates a complete virtual representation of it. This algorithm runs at 15Hz on a high-end NVidia GTX 580 graphics processing unit (GPU). Our tests of their software with an NVidia GeForce 9500 GT ran at 2Hz, which was too slow for the collaborative applications, which we are aiming for.

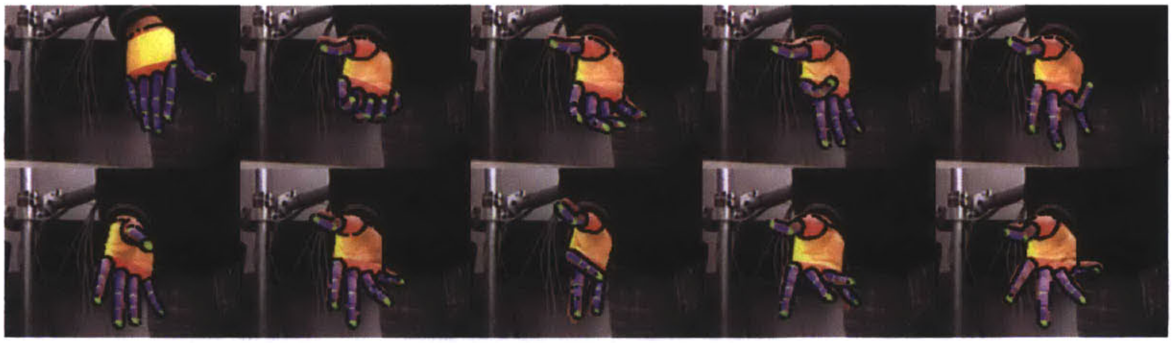


**Figure 20. The hand tracking algorithm presented in (Oikonomidis, Kyriazis, & Argyros, 2011) uses a single Kinect camera to track 27 DOF of the hand. It recreates a virtual representation of the hand but needs a higher-end GPU for real-time performance. Image courtesy of Oikonomidis, Kyriazis and Argyros.**

## 3. USER INTERACTION DESIGN

InReach consists of two setups which are remotely connected. For each of the two setups a Kinect depth sensor (Microsoft Kinect) stands on top of a large screen. Users are standing in front of this screen setup in a distance from 80cm to 3.5m. They can also move to the left and right. On the screen they see a 3D reflection of themselves as well as the reflection of their remote collaborators. All users are visible on the screen. In the virtual space they are standing next to each other (see Figure 21). Since the users are represented in 3D, they can move around in the virtual 3D space by physically moving around. They can stand in front of the other person or behind, or reach around an object.

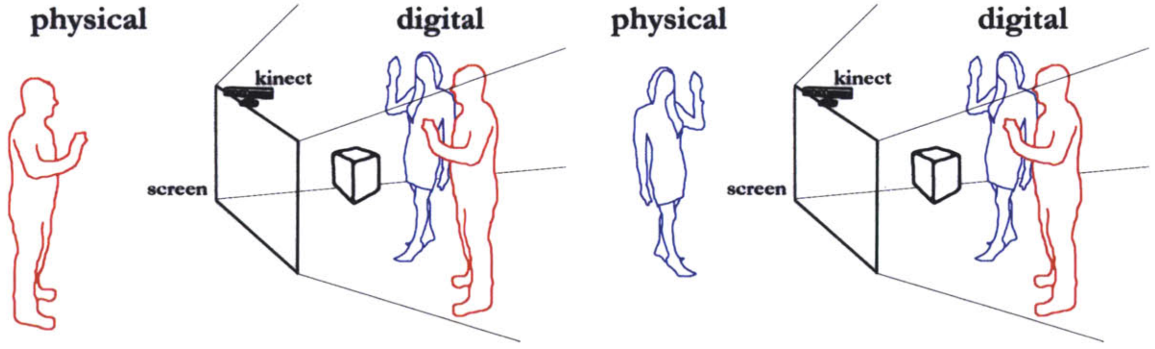


**Figure 21. InReach consists of two remote setups. One user is in one location the other is located somewhere else. Each user can move freely in front of the Kinect-screen-setup in his/her physical space. On screen in the digital environment both users, the remote and the local one, are visible. They are standing next to each other.**

To allow users to engage in the collaboration and be able to change the dataset they are inhabiting we developed different interaction metaphors which we will describe here.

### *3.1 Self-Representation*

In traditional video conferencing systems we see our conference partner in front of us in a face-to-face situation. In contrast to that we employ the metaphor of an augmented mirror. The user can see not only him/herself as in a normal mirror but also the remote collaborators in the same mirror, therefore augmenting the scene. HyperMirror (Morikawa & Maesako, 1998) is such a system for 2D. Morikawa et al. showed that in their HyperMirror system this kind of self-reflection could substitute for correct gaze. The project Reflection of Presence (Agamanolis & Bove, Jr., 1997) is also a mirror-like collaboration system. In this research the authors found that mirror-like systems are best accepted when users see themselves about life-size which we do for the InReach system. Instead of having a flat 2D representation, InReach creates a 3D representation of ourselves. In the context of virtual reality and animated agents such a setup has been presented in the ALIVE system (Maes, Darrel, Blumberg, & Pentland, 1997). In it a mirror view is augmented with virtual avatars with which the user can interact. In contrast to these systems, we augment the mirror with virtual objects as well as remote collaborators with the capability of manipulating the same space.

### *3.2 Manipulation*

Users are able to grab virtual objects in the 3D environment and translate, scale, and rotate them. Our system distinguishes one-handed and bimanual actions. Using one hand a user can only translate an object, with two hands s/he can translate an object, scale it and rotate it.

An open hand does not interact with the environment. Neither does a grabbing hand without touching an object. To move a virtual object a user needs to place their hand inside that object and grab; the object is then attached to their hand and can be placed somewhere else. A user can grab two different objects at the same time. If a user grabs one object with both of their hands s/he can also scale and rotate that object. The initial distance of the grabbing hands serves as relative measure. If the hands are moved closer to each other such that the relative distance becomes smaller, the object will be scaled down. If the hands move apart the object becomes bigger. To rotate an object the user has to place both hands inside the object as well. The hands define an axis, which the object is attached to. All of these interactions are illustrated in Figure 22.

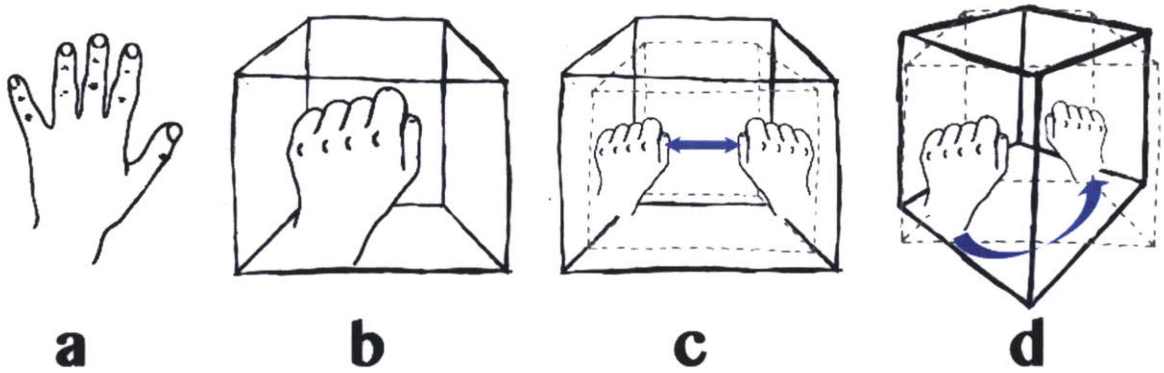


**Figure 22. (a) Open hand for no action. (b) Single hand grab to select object. Moving the hand will then translate the object. (c) Bimanual interaction for scaling and rotation. Scaling based on distance between hands. (d) Rotation based on axis defined by hands.**

In these interactions we do not distinguish the hands of local users from hands of remote users. This basically means that two users, be they locally together or remotely connected, can scale or rotate an object together. Each one of them needs to grab the object in question and the two hands define the scaling amount or the rotation axis.

### *3.3 Navigation*

To navigate in a virtual scene we use an adaptation of a travel technique called PenguFly (von Kapri, Rick, & Feiner, 2011). PenguFly was tested in a VR environment, the user's head and hands are tracked and the projected vector of head and both hands is used to define the direction of travel. See Figure 23.

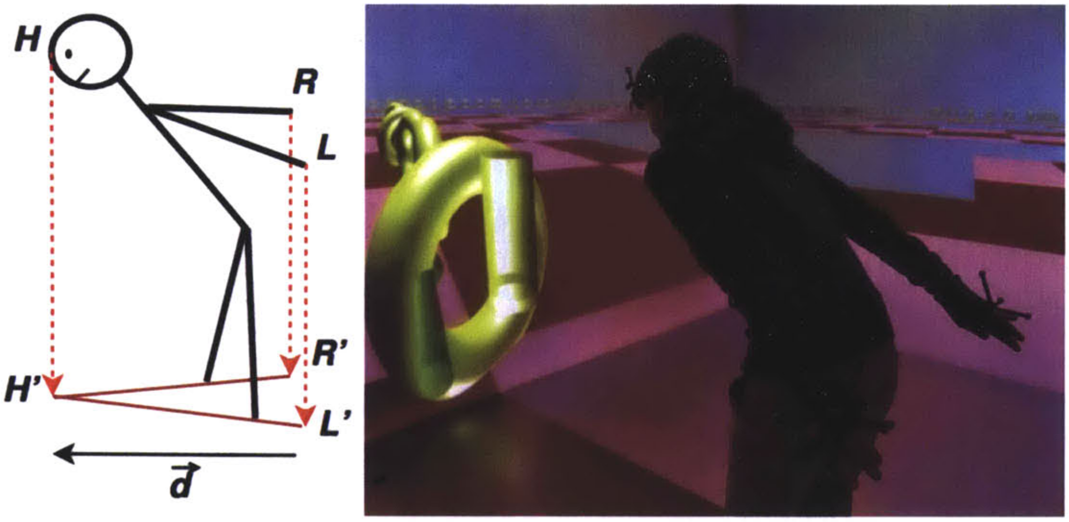


**Figure 23. PenguFly (von Kapri, Rick, & Feiner, 2011) is a body-directed travel technique. The standing user's head and hands are tracked. Their positions are projected onto the ground to define a 2D triangle. Travel direction is defined by d, whereas the velocity depends on the length of d.**

For our setup, instead of using the head and hands we track the torso of the user and take the leaning as an indication for the travel direction. We allow for a certain area around the user not to be classified as leaning, so that for example a simple head nod does not initiate flying around the scene. The speed rises exponentially dependent on the extent to which the user is leaning. We weigh each leaning direction differently since it is for example easier to lean forward than backward but we still want to accomplish the same speed for the same effort.

In a remote situation the user which is in the front will take control of the navigation if s/he is leaning towards a direction. Controlling the navigation basically means moving the virtual camera, the perspective onto the virtual scene. The other users follow that movement.

# 4. APPLICATIONS

Unlike many remote collaboration systems, InReach places all of its users in the same visual, virtual space. In a sense, each user becomes an 'actor' on the same 'stage.' Because of this theatrical element, InReach is useful for human interaction scenarios in which remote participants learn to handle specific, physical, real-world situations.

## *4.1 Interior Design*

One possible use case is for architects who are designing a house or the interior of a room and want to show their client the virtual model over a distance. Instead of using a remote desktop they could project themselves into the virtual space to get an impression how it would be like to live in that house and to give a virtual tour of the space. The assumption is that if a user sees her/himself standing in her/his future living room s/he would feel more present in that environment and could imagine what it is like to be there (see Figure 24). One advantage can be to have a better sense of the proportions.

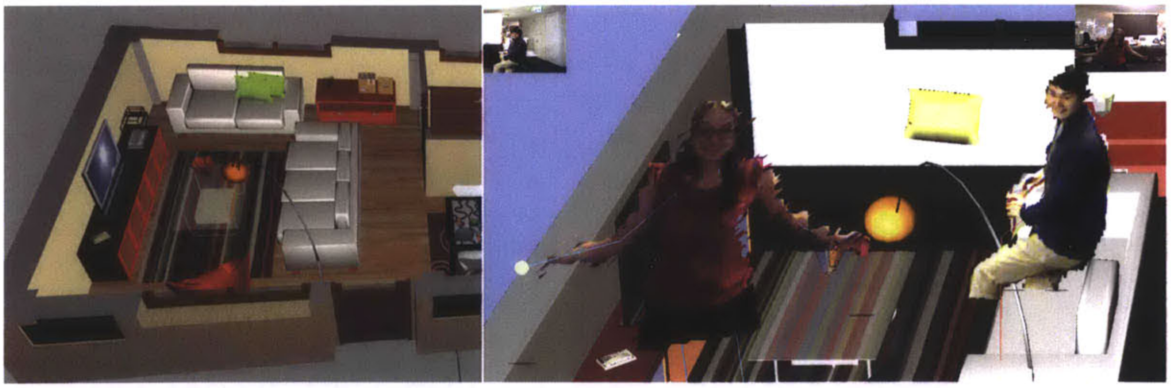


**Figure 24. (left) A virtual model of a living room. (right) Two remote users are projected into that living room. One person is pretending to sit on the couch.**

The architect can load in the model s/he designed and they could fly around the space and see if the spatial alignment of furniture is correct. For example they could test out if a cabinet is reachable by placing themselves there and playing out the action of opening the cabinet. Or a couple could visit their future home virtually together and play out certain scenarios, pretending to sit on the couch etc. Figure 25 shows how the setup would look like from one location.

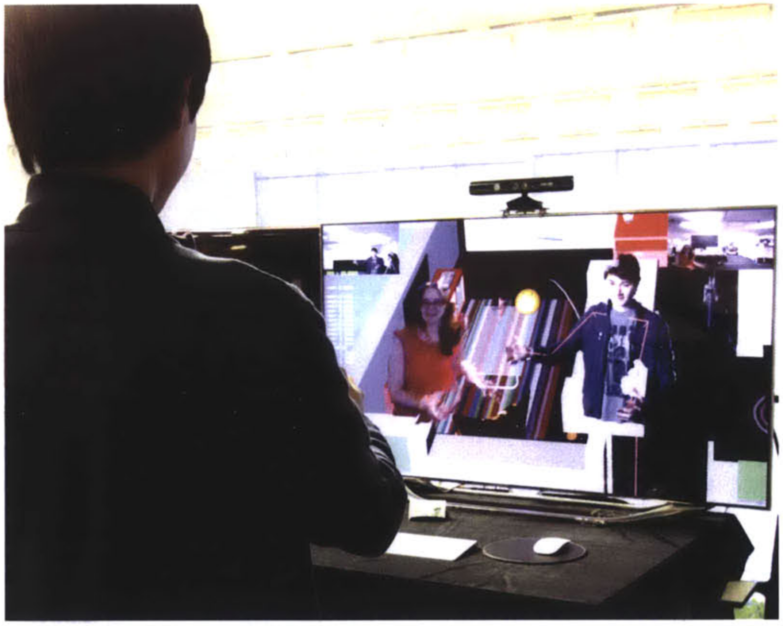


**Figure 25. This shows the setup of the system. The local user can be seen from the back and on the screen both users are visible standing in the virtual living room.**

Additionally users are able to move furniture around. For example if a living room table is not in the correct location, it could be moved somewhere else just by grabbing it (see Figure 26). Using two-handed interaction the rotational alignment of furniture can be changed as well.

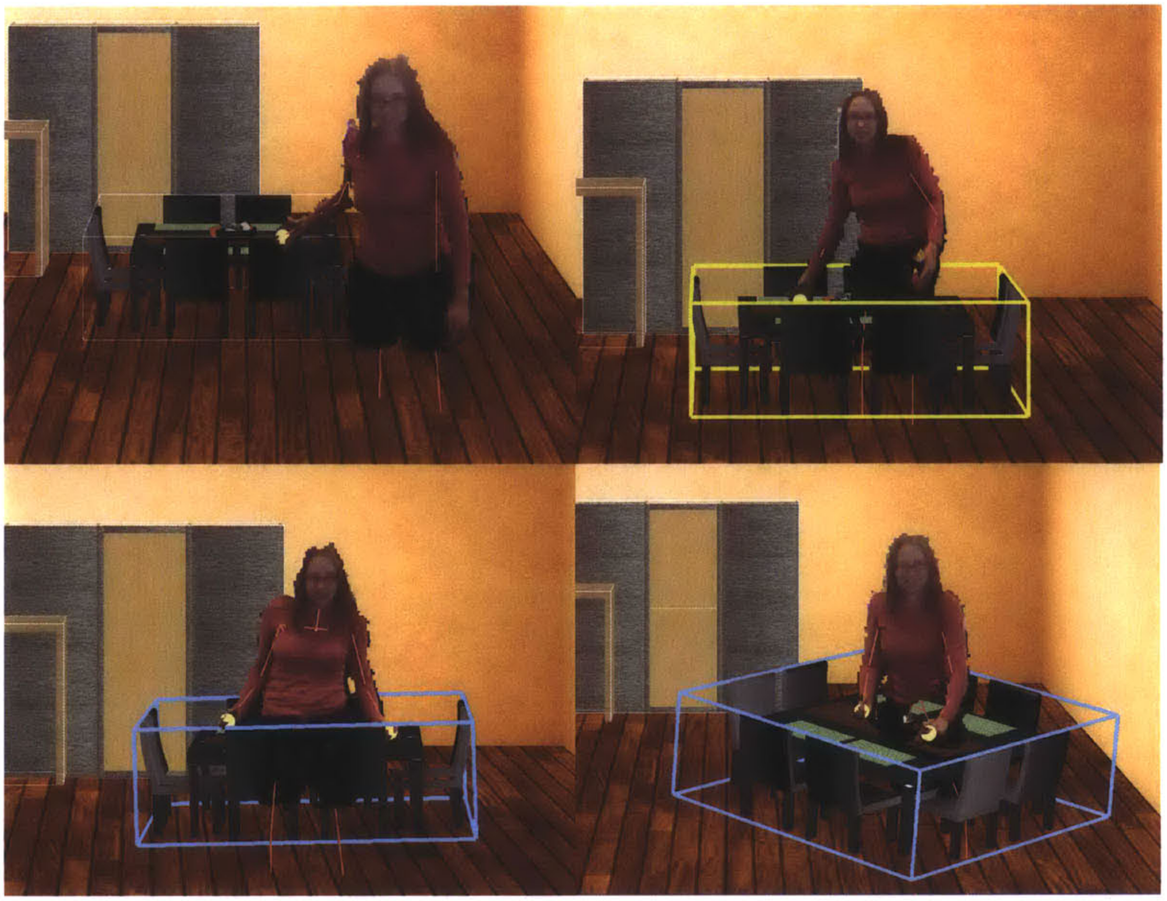


**Figure 26. In the interior design scenario a user can move around a room and place furniture somewhere else as well as rotate it around. Here you see a user grabbing a dining table and moving it to a different location in the room as well as turning it.**

### *4.2 Visiting cities virtually with your friends*

Another implemented application is for virtual tourism or in general flying around a 3D model. We designed a simplified model of the city of Munich in which a user can project her/himself. By leaning in a desired direction s/he can move around the virtual space (see Figure 27).

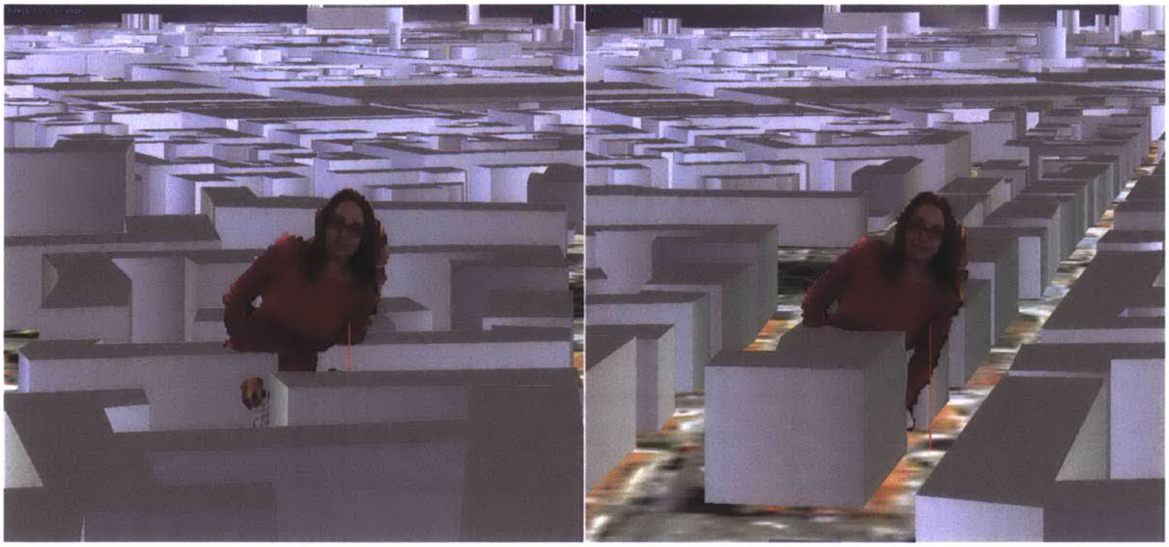


**Figure 27. A user is flying around a simplified virtual model of the city of Munich by leaning into the direction s/he wants to fly to.**

In Figure 28 the local as well as the remote user are sharing the virtual space. The front user is taking control of the navigation, whereas the back user always follows along and explores the city through the direction of the other one.

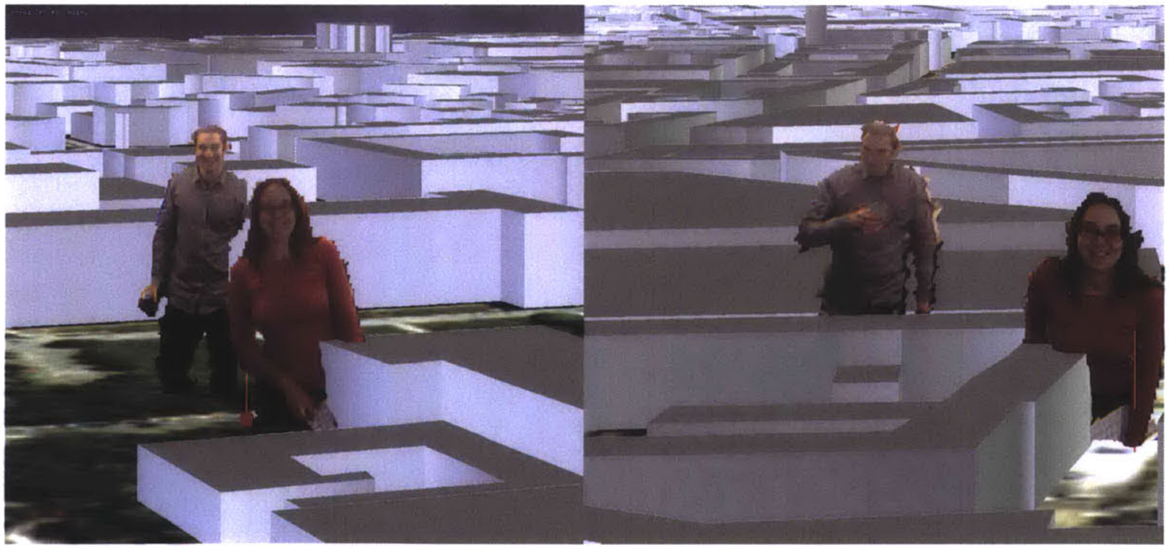


**Figure 28. Two users are exploring the virtual city together. The front user takes control of the navigation.**

### *4.3 Studying 3D structures*

As a conceptual application we looked into using InReach in an educational setting. FoldIt (FoldIt, 2013) is a protein folding game which allows a player to learn about protein structures and their shapes. These influence their function and behavior. In FoldIt a player can drag the backbone of the protein or sidechains and move them either closer together to fill empty space or farther apart to reduce clashes. Figure 29 illustrates how this FoldIt game would look like in an InReach remote setup. The teacher and the student are standing next to the protein structure that they are inspecting. The teacher could point to certain parts of the protein to explain the structure of the protein such as the backbone and the sidechains. They could grab a sidechain to move it closer or farther away from the backbone and see how this influences the shape. This mirrored view of the student seeing her/himself touching and interacting with the protein could create a direct understanding of the spatial structure of such proteins.

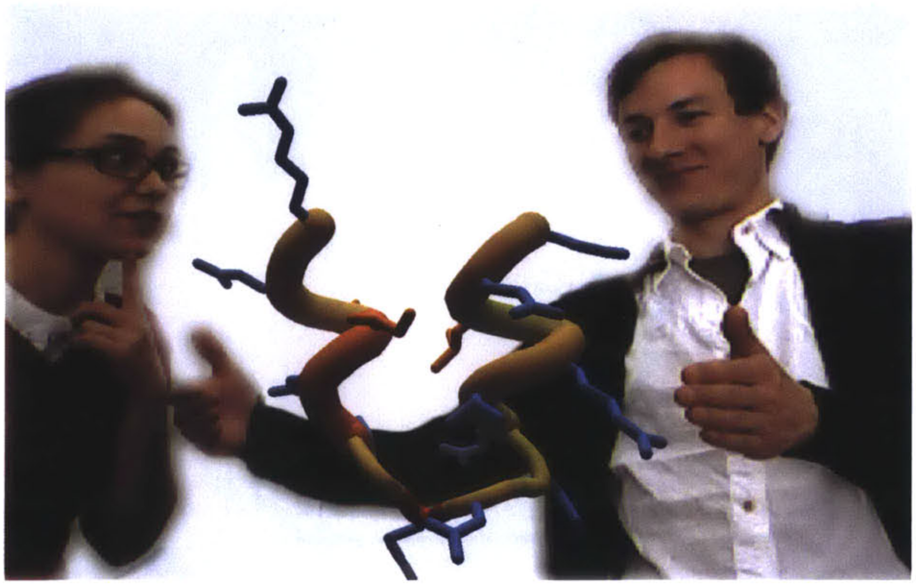


**Figure 29. A conceptual application focuses on an educational game in which a teacher and her/his students can "inhabit" a protein and point out particular components and features.**

## 5. IMPLEMENTATION

The InReach system consists of two remote stations connected over the network. Each station consists of a 3D stereoscopic screen as well as a Kinect depth sensor. Both systems run on an Intel Core i7 processor and an AMD Radeon HD 6630M GPU. The individual building blocks of the system, Sensing, Tracking, Hand Pose Detection, 3D Manipulation, Physics as well as Rendering, are described in the following sections (see Figure 30).

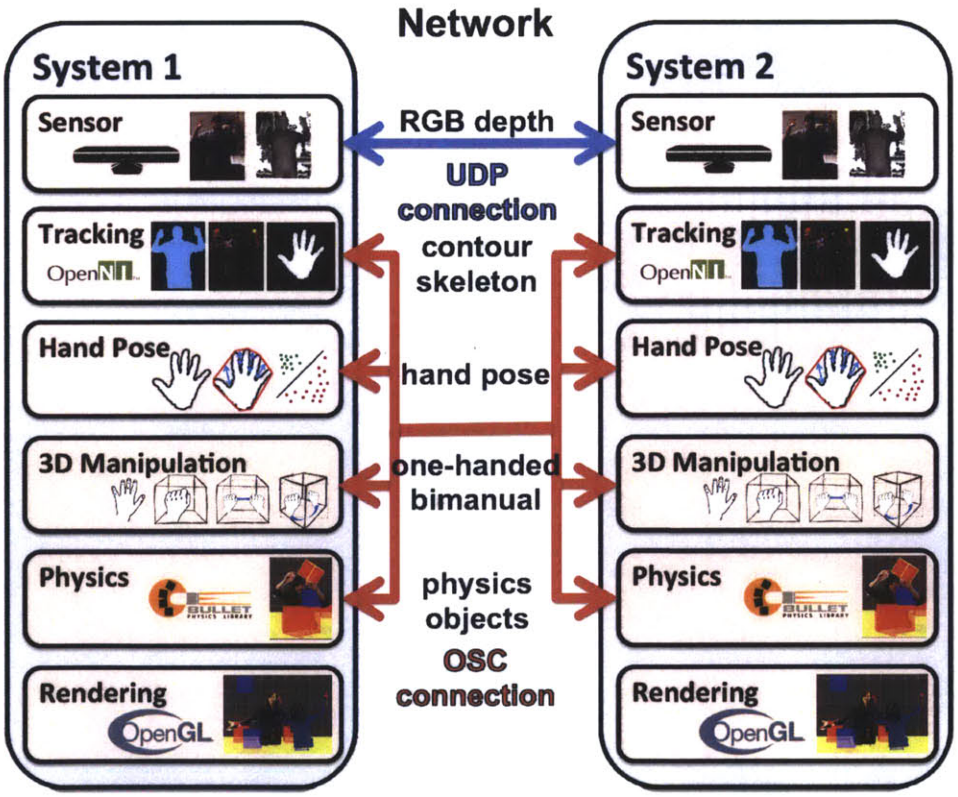
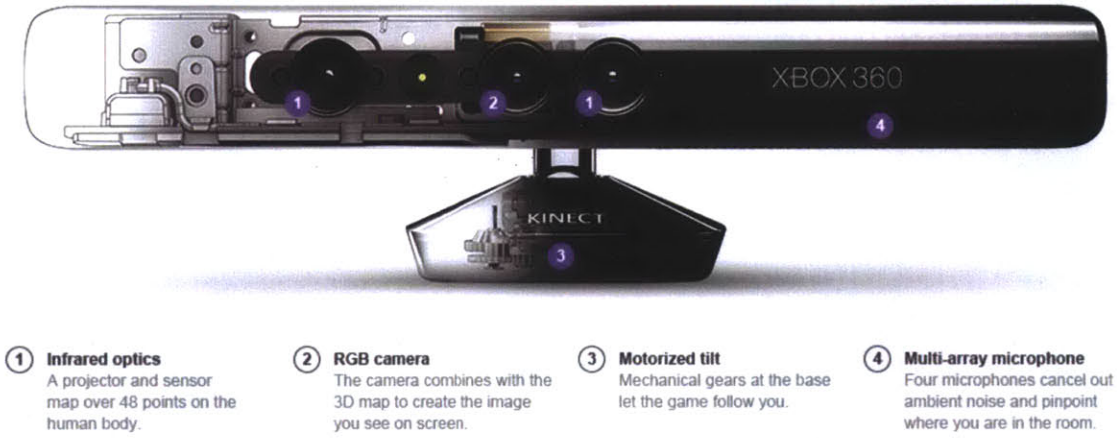


**Figure 30. This diagram shows the architecture of the InReach framework. Two stations are connected over the network; a UDP connection is set up to send the RGB and depth images, and the OSC connection transfers meta data such as user and skeleton information, hand poses and hand actions as well as states of virtual objects over the network.**

### *5.1 Sensor and Tracking*

The Microsoft Kinect (see Figure 31) is a sensor that captures an RGB and depth image of 640x480 in resolution. The capturing rate is at 30 frames per second (FPS). The viewing angle is 43° in the vertical axis and 57° in the horizontal. The Kinect has a depth sensing range of approximately 80cm-3.5m. This gives a large area of interaction. The Kinect has a motor which allows to change the viewing direction in the vertical axis by ±28°. Additionally, it has a four-microphone array to detect the direction from which sound is coming.



(1) **Infrared optics** A projector and sensor map over 48 points on the human body.

(2) **RGB camera** The camera combines with the 3D map to create the image you see on screen.

(3) **Motorized tilt** Mechanical gears at the base let the game follow you.

(4) **Multi-array microphone** Four microphones cancel out ambient noise and pinpoint where you are in the room.

**Figure 31. The Microsoft Kinect sensor captures a color as well as a depth image in the resolution of 640x480. Image courtesy of Microsoft.**

Figure 32 shows an example of a colored image captured from the Kinect sensor as well as its corresponding depth image. For our application we are mirroring the images to represent a mirror view of oneself.

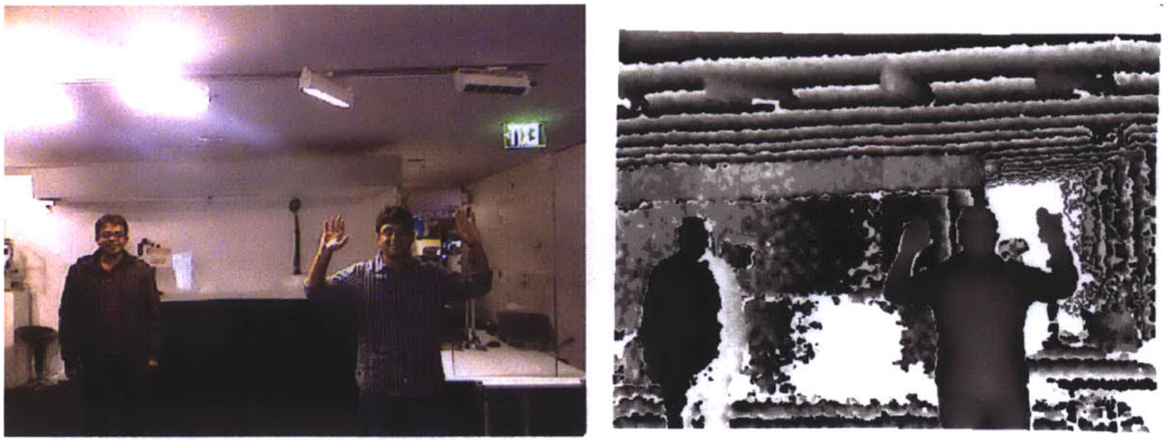


**Figure 32. The Kinect captures the color image (left) every frame as well as the depth image (right) at a rate of 30 FPS.**

The OpenNI framework (OpenNI) is an open source SDK used for the development of 3D sensing middleware libraries and applications. OpenNI allows to interface with the Kinect hardware and retrieve the depth as well as RGB pixels. NiTE (NiTE) is a perception algorithms layer which enables to perform functions such as hand locating and tracking, scene analyzer (separation of users from the background) and accurate user skeleton joint tracking. Using OpenNI and NiTE we can assign pixels to different users at the same site, therefore distinguishing users from each other and from the background (see Figure 33 (left)). These user masks are used to extract the correct color pixels from the captured RGB video and combined with the depth values as given back in the real-world coordinate system. Each point in space has an assigned color from which we generate a 3D mesh for each user. We can distinguish up to eight different users at each remote location. Only knowing the user's mesh in 3D space is not enough to allow for interactions. Therefore, we are tracking each user's skeleton. After the user stands in an initial tracking pose (with both arms angled up, as shown in Figure 33 right)) the system can detect where his/her hands are, where the head is, the torso etc. This initial pose provides the system with the calibration data, it starts learning the dimensions of the user (the length of the limbs, the user height, etc.).

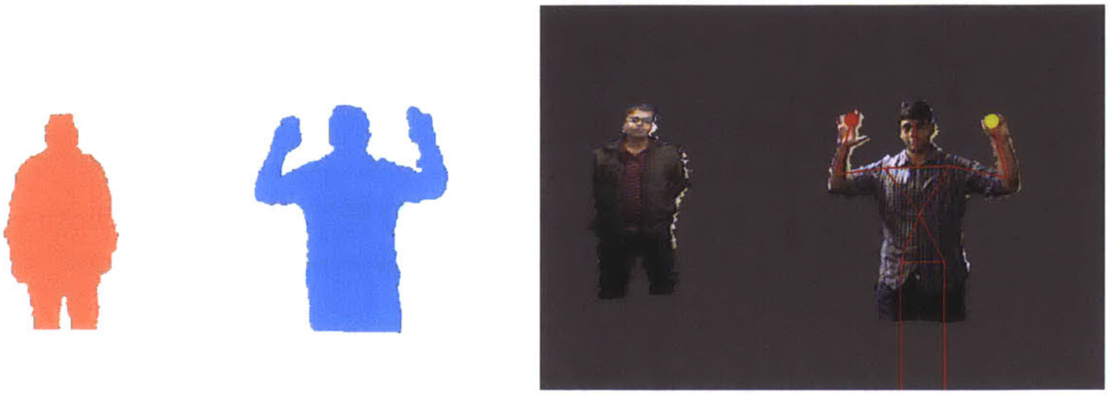


**Figure 33. (left) Different users are distinguished based on the depth image. (right) The 3D meshes of the users are generated from the color and depth image. If the user goes into an initial tracking pose his/her skeleton will be tracked consequently.**

Additionally, we need to have an active input capability for which we need to detect the hand poses of each user, this will allow her/him to actively grab or release virtual objects. We first extract the user's hand mask. For this we use the hand positions which were previously calculated by OpenNI's skeleton tracking system. We only analyze depth pixels from that particular user's depth mask and take only depth values in a circular area around the hands. Since these hand mask calculations are happening on a 2D image, this area needs to be proportional to the distance of the hand from the camera. To handle cases where the user's hand is in front of the body we discard depth values that are out of a 6cm z-axis range around the hand. A representation of these steps is illustrated in Figure 34.

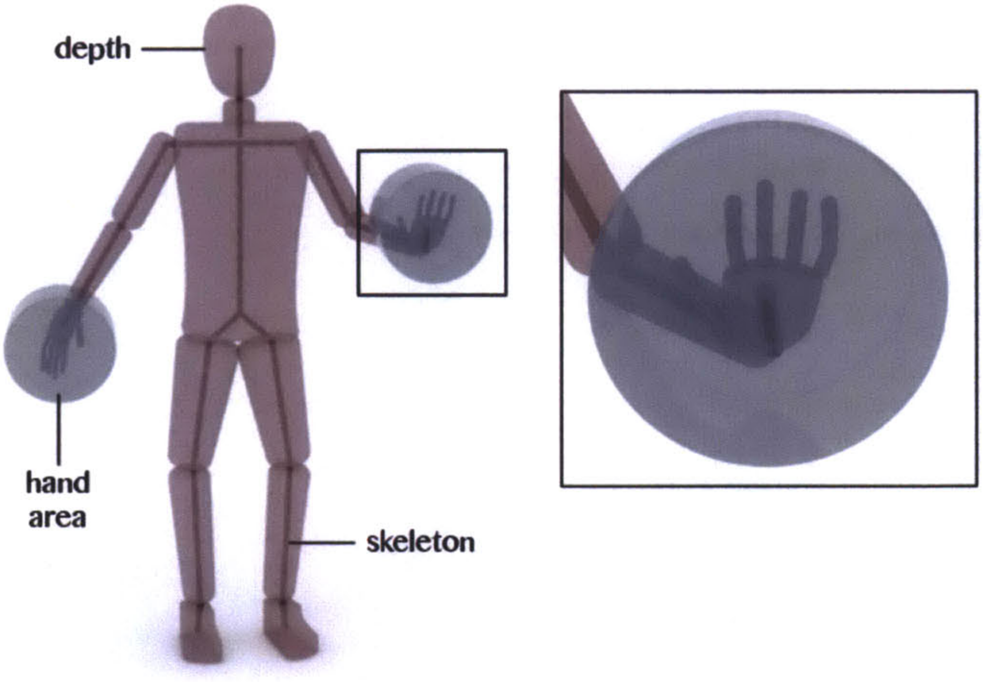


**Figure 34. To extract the user's hand mask only depth values around an area of the skeleton's (x, y, z)-hand-position are considered.**

## *5.2 Hand Pose Detection*

In order to detect if a user is touching an object we use an algorithm which uses a 3DOF plus action hand pose detection. We distinguish hand gesture from hand pose; a gesture is an action in time and a pose is detected per time instant.

Other systems have approached hand manipulation by implementing the detection of a pinching pose (Wang, Paris, & Popovic, 2011) (Wilson A. D., 2006), which can be recognized quite robustly in setups where the camera is positioned above the user and looks down onto the scene. In a setup where the camera faces the user, we would have to ensure that the camera always sees the hole in the hand, making this pose less suitable for our setup.

We decided to recognize two different hand poses: open hand (Figure 35-1) and closed hand (Figure 35-2). The closed hand is not ideal for precise interactions but its detection is quite robust. We use OpenCV to calculate the contour (Figure 35 b) based on the previously computed hand mask. The contour of the hand has different characteristics based on the hand pose for which we define two properties: (1) maximal normalized convexity defect (see Figure 35 c) and (2) ratio of height to width of best fitting ellipse (see Figure 35 d). Hands with high convexity defect are classified as open, and hands with low convexity defect are classified as closed. The convexity defect is the maximal distance between the convex hull and the contour itself.

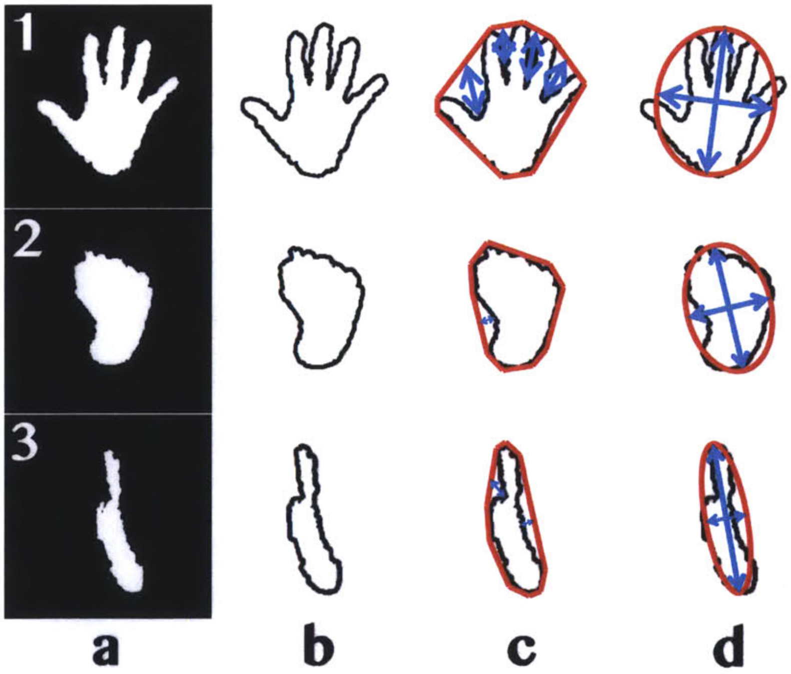


**Figure 35. (1) Open hand. (2) Closed hand. (3) Open hand sideways. (a) Hand mask. (b) Hand contour. (c) Convexity defect. (d) Minimal fitting ellipse.**

To overcome recognition problems that arise from holding an open hand sideways, which does not result in high convexity defect we introduce a third class for open hand sideways (see Figure 35-3). A hand is classified as such if it has small convexity defects and a higher ratio of the width and length of the minimal fitting ellipse. See Appendix B.1 Hand pose detection for the C++ implementation.

These properties of a training set are fed to an adaptive naïve Bayes classifier (ANBC) introduced by Gillian et al. (Gillian, Knapp, & O'Modhrain, 2011). This allows us to train the system with differently sized hands, thereby making it more robust. See Appendix B.2 Setting up the adaptive naïve Bayes classifier for code samples.

Another limitation is that due to blur, fast movements are not always correctly classified and the user is likely to drop an object they are grasping. Simple time filtering with a window of 90ms makes the results more stable.

### *5.3 Physics*

We included physics in our virtual environment using the Bullet physics library (Bullet physics library). In our current implementation only virtual objects can interact with each other. Releasing objects in mid-air will let them fall to the ground. A user can push objects away if he is holding another object (see Figure 36). For the user mesh to interact with the virtual objects we would need to give the mesh physical characteristics and define it as Bullet physics shape.

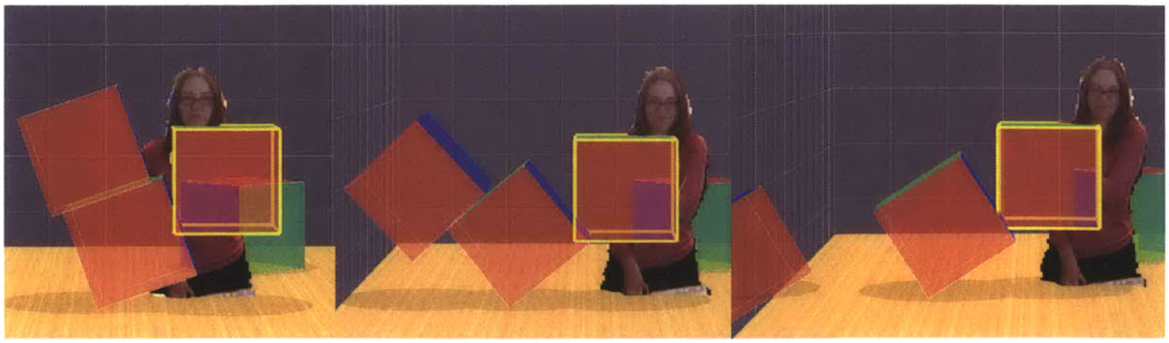


**Figure 36. A user is grabbing a cube and pushing away two other ones. This sequence shows the effect of the physics.**

### *5.4 Rendering*

The 3D environment is rendered using OpenGL. Each user is represented by a 3D mesh based on his/her depth and color values (see Figure 33 (right)). The system runs on conventional non-stereoscopic displays as well as on stereoscopic displays. Stereoscopy is achieved by rendering the scene twice using parallel axis asymmetric frusta (Figure 37 (left)) in contrast to using regular perspective projection frusta (Figure 37 (right)). For implementation details please refer to (Kooima, 2009).

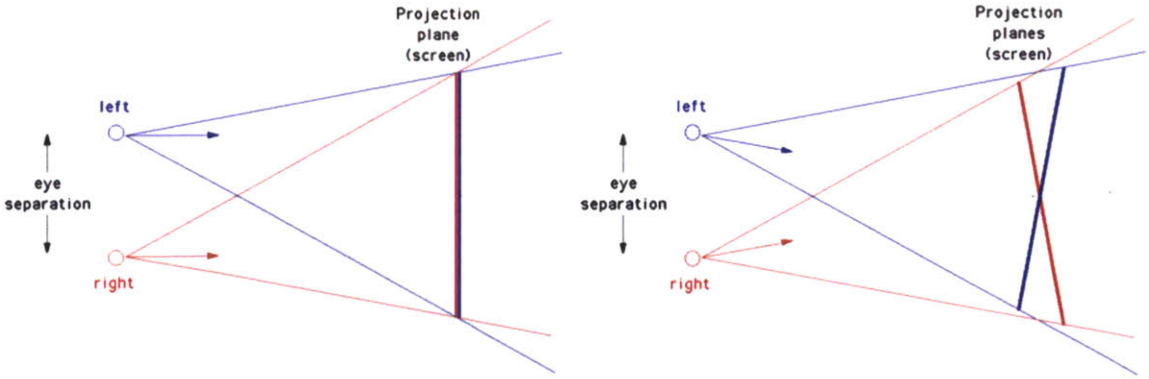


**Figure 37. (left) Using parallel axis asymmetric frusta for the stereoscopic rendering aligns the projection planes for the left and right eye on top of each other. (right) Regular perspective projection frusta would not overlay the projections planes of the left and right eye in a stereoscopic setup.**

Eye distance is set to 7cm. The perspective of the scene is continuously updated based on the position of the tracked user's head. Stereoscopic rendering gives better cues of where the body and hands are in the virtual space, making it easier to grab objects and see where you are placing them. Moreover, shadows of the hands and objects are projected onto the floor, which make it easier for users to determine their hands' positions in relation to the objects.

### *5.5 Network Protocol*

The system is set up with a peer-to-peer connection. Two data channels are used: a user datagram protocol (UDP) channel, which is used to send the compressed RGB and depth images over the network and a Meta data channel, which uses Open Sound Control (OSC) as content format. To avoid unnecessary recalculation, we send user contours, skeletons, and hand poses, which have been calculated locally, over the OSC channel. The video channels are compressed and decompressed with the open source Turbo JPEG library (Libjpeg-turbo).

## 6. USER STUDY AND EVALUATION

The InReach system has been used by many researchers as well as visitors at our laboratory for the past 9 months. We found that people get excited when they see themselves in the system. It was also interesting to see that as a first reaction without introducing the system they thought the remote users were previously captured instead of remote real-time collaborators. As soon as they understood that these people were present in a different location they started interacting. As for the interaction with virtual objects, users found it magical that they could grab a virtual object with just their hands. This broke down the barrier between the digital and the physical. Users liked to grab an object and scale it up so that its resolution is larger than the virtual room. They stated feeling more powerful probably because this action is not possible in reality. Throwing these digital objects around and seeing how they fall to the ground, bounce up or hit other objects was fascinating because their own actions influenced the way the objects behaved.

In order to do a more systematic evaluation, we designed a user study to test the feeling of co-presence for different arrangements of the collaborators and levels-of-detail of the self-representation. Co-presence was used and measured as described in (Slater, Sadagic, Usoh, & Schroeder, 2000). The experiment was a within-subject design and the independent variable was the self-representation of the local user. Users were asked to correctly sort colored cubes into colored zones. The main hypothesis was that participants would have a stronger feeling of co-presence in the augmented mirror/side-by-side setting compared to the control conditions.

### *6.1 Participants*

We recruited ten unpaid participants (4 females, 6 males, mean age = 27.1, SD = 4.2) through email. All were able to walk unassisted and had full use of both arms. Only one participant stated he used the Kinect/Wii twice a week, two participants used it weekly, two monthly and five once a year. Since we tested for co-presence, two subjects participated per experiment. For each session, we generated a random order of the following three conditions.

### *6.2 Conditions*

The three conditions differ in the self-representation aspect of the local user. The representation of the remote participant is the same in all conditions. The remote user is represented as a 3D colored mesh generated from the Kinect RGB and depth values.

#### *Face-to-Face (F2F)*

The F2F representation mode is the more traditional approach where the users are facing each other in a virtual environment. A local user can only see their hands represented as virtual circles. See Figure 38.

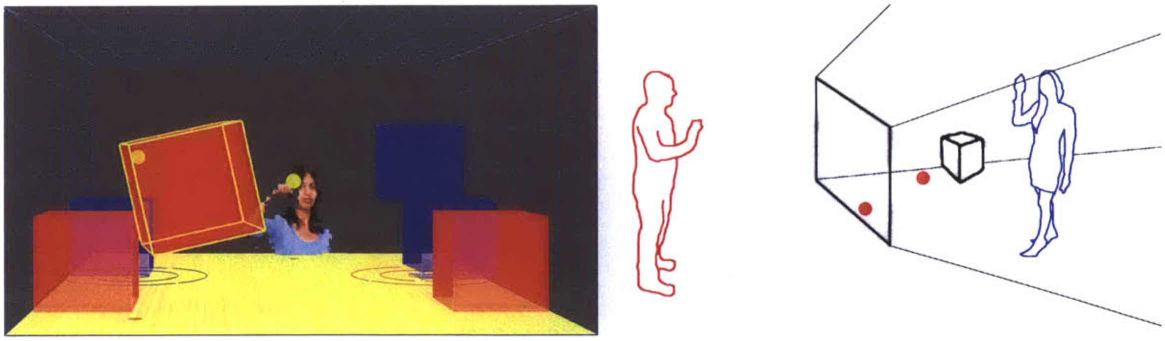


**Figure 38. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Face-to-Face (F2F) setup the local user can only see his/her hands as circles in the environment. (left) Actual view. (right) F2F setup.**

### *Desktop VR (DVR)*

Similar to the previous condition, in DVR users are facing each other in the virtual space, but they are represented as whole skeletons seen from their backs instead of mere spheres. In a way, the user animates a virtual avatar in this representation. See Figure 39.

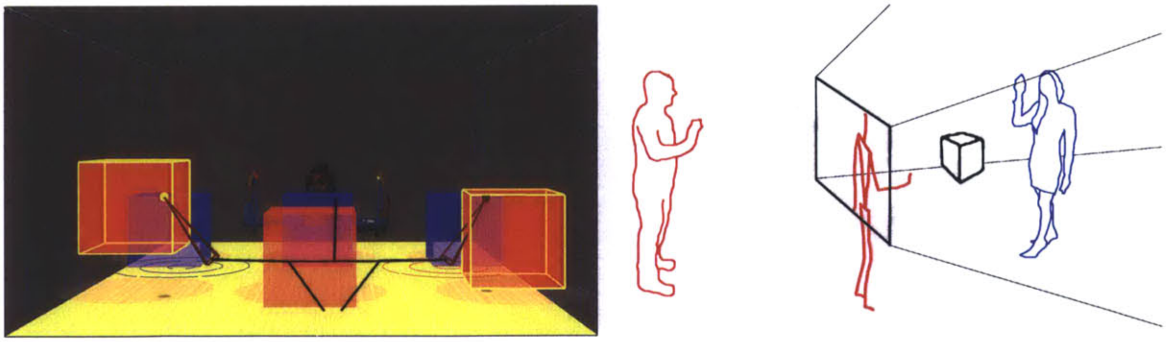


**Figure 39. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Desktop VR (DVR) setup the local user can see his/her skeleton from the back. The arms are red and the rest of the body is gray. (left) Actual view. (right) DVR setup.**

***Side-by-Side (SbS)***

The SbS condition is the InReach interface, which has already been described in the Section 3.1 Self-Representation. Local users are mirrored and their 3D meshes are displayed next to those of the remote user. Both users cohabit the same virtual space next to each other. See Figure 40.

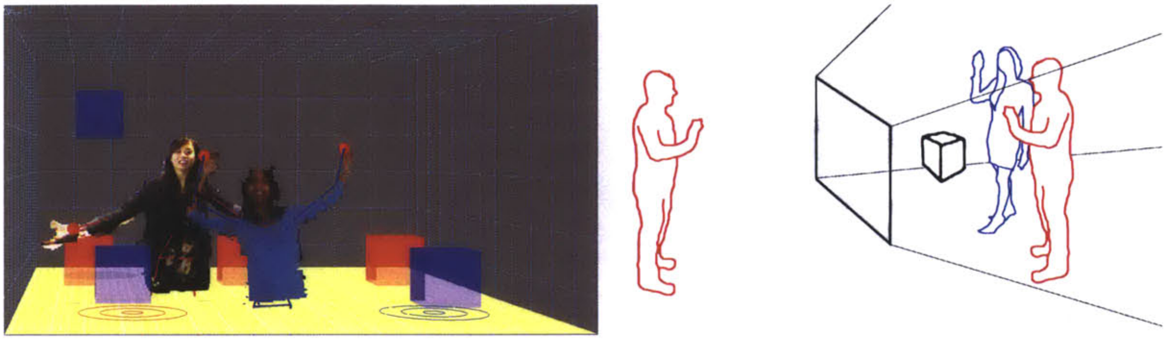


**Figure 40. The local user is represented in red, the remote user in blue. The remote user is represented as a 3D mesh. In the Side-by-Side (SbS) setup the local user sees a mirrored 3D mesh of him/herself. (left) Actual view. (right) SbS setup.**

### *6.3 Setup/Equipment*

For the experiments we used two remote setups in different rooms. Both rooms were far apart so that no participant could see what the other participant was doing except through the screen. We used two 3D stereoscopic screens. The audio connection was done through Skype.

One setup had a Samsung SmartTV LED 8000 with 54.6 inch screen diagonal and 1920x1080 resolution. It used active shutter technology for the 3D stereo effect. The user had to wear shutter glasses. The Kinect was on top of the SmartTV at 1.58m height with an angle parallel to the ground. To not get distracted by incoming light from the background and clutter in the environment we attached black cloth behind the screen which allowed for a better stereo effect.

The other setup used an LG 42LM6200 TV with 42inch screen diagonal and 1920x1080 resolution. It had passive stereo technology with circular polarization. The user had to wear glasses. The Kinect was on top of the LG TV at a height of 1.6m with an angle parallel to the ground. We also attached black cloth behind the screen in this setup. Both machines used an Intel Core i7 processor and an AMD Radeon HD 6630M GPU. In stereo mode the application ran at 25 FPS.

### *6.4 Experiment Procedures*

First, we introduced participants who did not know each other. Then, we gave a brief introduction of the system and talked about the 3D manipulation and how to respond if the system had problems recognizing the hand pose. After that, one investigator took one participant to the other room. The experiment was divided in three sessions, one for each interface. The ordering of the interfaces was randomized. Each session started with a training phase and continued with five sorting tasks. After each training phase the participants were instructed to work collaboratively on the task. After each session the participants were asked to fill out a co-presence questionnaire. When they had seen all interfaces, participants were asked to fill out the rest of the questionnaire. And after that a short interview followed. In total the experiment lasted 35-45 minutes.

During the training phase participants were asked to get accustomed with the system. There was no goal. Participants could grab objects from a menu and place them in the environment. They experimented with translating the objects as well as scaling and rotating them. As soon as they felt comfortable the investigator switched to the sorting task.

The task consisted of sorting three blue and three red colored cubes which were placed in the environment, onto corresponding colored spots. The cubes disappeared as soon as they were in a close proximity to either of the two spots. There were five such scenes which differed in the placement of the cubes. Some scenes were more difficult than others.

### *6.5 Measures*

As subjective measures we assessed co-presence as well as frustration and intuitiveness of the 3D manipulation. To measure co-presence we used the questions introduced by Slater et al. (Slater, Sadagic, Usoh, & Schroeder, 2000):

1. In the remote meeting to what extent did you have the sense of being together with the other person?
2. Continue to think back about the remote meeting. To what extent can you imagine yourself being now with the other person in that room?
3. Please rate how closely your sense of being together with the other person in a real-world setting resembles your sense of being with them in the virtual room.

We used two 5-point Likert scale questions for the 3D manipulation, which were:

1. On a scale from 1-5 (1= least frustrating, 5= most frustrating) how frustrating did you experience the 3D manipulation?
2. How intuitive do you think the 3D manipulation is (1-least intuitive, 5 most intuitive)?

At the end of the experiment we asked the participants to rank the interfaces, first regarding sense of co-presence and second regarding game play. Please see APPENDIX A: QUESTIONNAIRE for reference.

### *6.6 Evaluation*

The frustration experienced with the 3D manipulation had a mean of 2.2 (sd=1, median=2, mode=none) across all conditions, where 1 was least frustrating and 5 most frustrating. The measure for intuitiveness of the manipulation resulted in a mean of 4.2 (sd=0.6, median=4, mode=4) across all conditions (1=least intuitive, 5=most intuitive). In general participants could not perform fast movements while holding an object without losing it. This was considered frustrating in some cases. The intuitiveness measure showed that it was in fact intuitive for participants to understand the 3D manipulation across all conditions. See Figure 41.

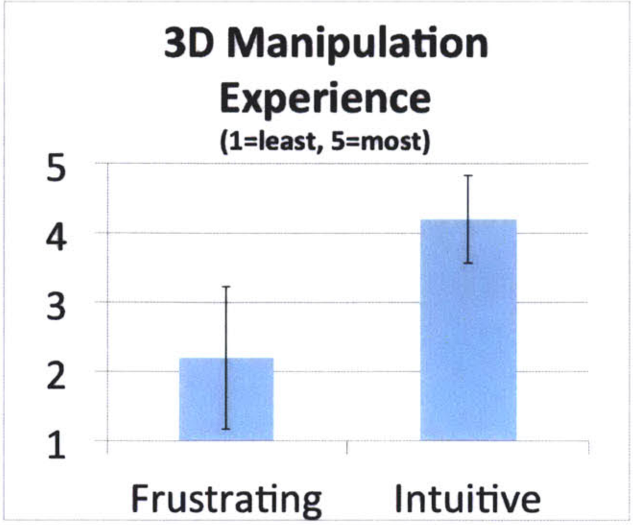


**Figure 41. Subjective rating of frustration and intuitiveness for the 3D manipulation experience over all conditions (error bars show standard deviation).**

For the sense of co-presence ranking, five participants favored the F2F condition. Three participants favored the SbS condition and only two the DVR condition. There were similar results for the ranking regarding game play with five participants ranking F2F first, two SbS, and two DVR. See Figure 42.

The three one-way within subject analysis of variance (ANOVA) across all interfaces for the three co-presence questions revealed no significance (1. Question: F(2,27)=0.045, p=.96; 2. Question: F(2,27)=0.329, p=.72; and 3. Question F(2,27)=0.187, p=.83.). For the statistical analysis we used the free software R (R Core Team, 2012).

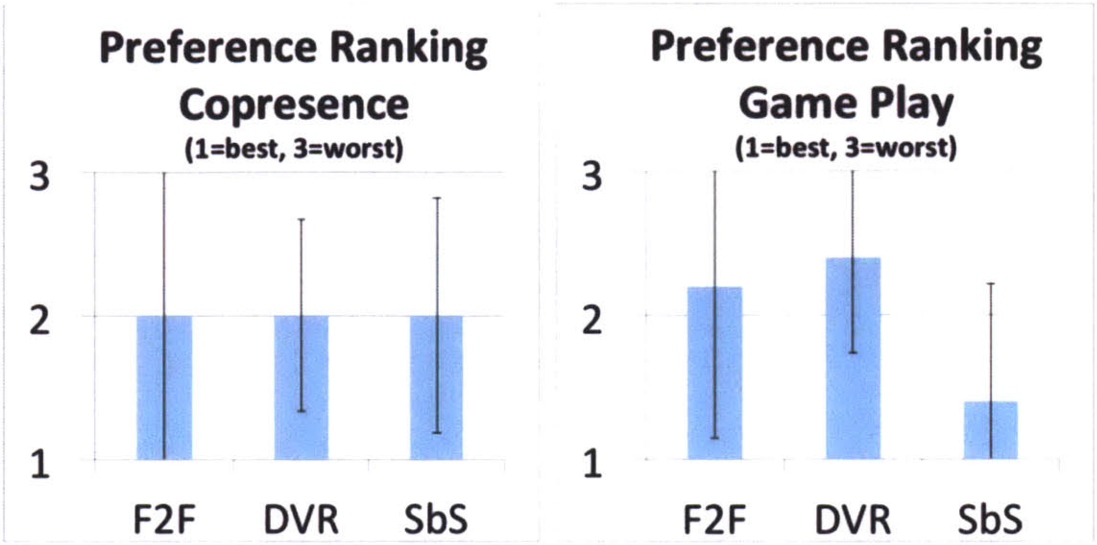


**Figure 42. (left) Subjective preferences of conditions in terms of co-presence ranking. The three conditions are Face-to-Face (F2F), Desktop VR (DVR), and Side-by-Side (SbS). (right) Subjective preferences of conditions in terms of game play rating. (Error bars show standard deviation.)**

We believe these results are due to the nature of the questions of the co-presence questionnaire and the definition of co-presence itself. Co-presence is the feeling of being here in the real world and having an interaction with a person that is face-to-face. The condition F2F was designed to be the one that tries to imitate reality and therefore gets better results on the co-presence questionnaire. Still, these differences are not significant. The five participants who favored the F2F condition gave as reason that it allowed them to communicate in a natural way. It was easy for them to dive into the environment since they did not see themselves. One participant stated that she "could focus on the other person" better.

It is interesting to see that the three participants who favored SbS did not compare it to reality had a feeling of being with that person. They could relate themselves to the environment the most and feel within. One participant said "if I see my body in the room, I feel I am inside". It gives "more personal interaction possibility". Another participant compared this to dreams, where he sometimes sees himself in a third-person perspective, which therefore felt very familiar. Yet another participant felt that SbS was "neat because [she] could give the other person a hug because [she] could see her arms".

The two participants who favored the DVR condition gave different reasons for their choice. One liked that the zones of interaction were implicitly defined for each person plus a zone in the middle, which was for collaboration. The other person knew where her body was in the space; she took advantage of proprioception. However, that particular participant felt more co-presence for SbS than F2F but could not get used to moving around in the space.

In summary, it seems that the F2F condition felt most like reality, a disadvantage is that there might not be enough feedback on the position of one's own body in the environment, which makes it more difficult to interact with. The DVR condition felt most effective in terms of the sorting task even though users might focus more on the skeleton than on the partner. The SbS condition was better in terms of relating oneself to the collaborator and the environment and creating a shared atmosphere. The main disadvantage was that participants could not as easily rely on proprioception. They were looking into a mirror, which felt confusing when trying to grab an object. They found themselves completely reliant on visual feedback.

Most of the participants imagined interactions with the other person's body. They wanted to be able to collide with the other or push him away. Also audio feedback when touching such as when giving a high-five would enhance the experience. Participants wanted to interact with the whole body in the environment such as using the limbs to kick objects around. One participant was missing precise control.

It feels like the SbS condition is much better in remote collaboration settings that focus on the interpersonal space in a playful way. In SbS you can step on each other's toes but can relate to each other and play with the representation of the collaborator. Collaborators are doing things they would probably not that easily do in a real world setting. The F2F and the DVR condition have a clear boundary of personal spaces. This seems to put the focus on the shared workspace. On the other hand, considering that the task was not collaborative in itself we could argue that a task that required more coordination with the other person would favor SbS over F2F. As an example we are thinking of a person holding a nail while the other person pounds on it with a hammer, or a task of lifting something together.

## 7. CONCLUSIONS AND FUTURE WORK

We presented InReach, a 3D remote collaboration system that allows multiple users to be projected into a shared virtual space where they can inhabit 3D models and manipulate 3D objects with their bare hands. Our work contributes a novel implementation, which combines 3D capture and replay through a Kinect depth sensor, correct 3D perspective view, freehand interactions and a remote setup for correct physics calculations. In addition to the system, we contribute an experiment that analyzes the effect of differences in self-representation on the feeling of co-presence.

We are encouraged by the initial feedback from users of InReach and are working to upgrade to more precise hardware, improve and evaluate gestural capabilities, simulate more realistic physics interactions, and study our system in a variety of industry contexts. In a future version of the system we would like to include better recognition of the hands with techniques presented in (Wang, Paris, & Popovic, 2011) (Oikonomidis, Kyriazis, & Argyros, 2011). This would allow for precise detection of finger articulation and orientation in space. We would also like to include additional sensors with finer grained depth capability such as LeapMotion (Leap Motion) or the CREATIVE* Interactive Gesture Camera Developer Kit (CREATIVE* Interactive Gesture Camera Developer Kit).

To support mode switching in collaborative scenarios and the transfer of control, we would like to add recognition of an additional set of gestures using a Support Vector Machine. We also would like to include interactions that utilize the whole body such as kicking or pushing blocks out of the way. This could be accomplished by adding physics that include friction and optical flow between a model of the body and the virtual objects in the scene as outlined in Andrew Wilson's research (Wilson, Izadi, Hilliges, Garcia-Mendoza, & Kirk, 2008).

Our long-term goal is to study how our system could be integrated in industries that use 3D models extensively in their design processes. These include fields such as industrial design, online education, game design, architecture, and software visualization.

# Bibliography

Agamanolis, S., & Bove, Jr., V. (1997). Multilevel Scripting for Responsive Multimedia. *MultiMedia , 4* (4), 40-50.

Benko, H., Jota, R., & Wilson, A. (2012). MirageTable: freehand interaction on a projected augmented reality tabletop. *SIGCHI Conference on Human Factors in Computing Systems* (pp. 199-208). Austin, Texas, USA: ACM.

*Bird by Leonar3Do*. Retrieved May 9, 2013, from http://leonar3do.com/

*Bullet physics library*. Retrieved April 21, 2013, from http://bulletphysics.org

*Cisco TelePresence*. Retrieved April 20, 2013, from http://cisco.com/en/US/products/ps7060

*Cisco webex*. Retrieved April 20, 2013, from http://webex.com

*CREATIVE* Interactive Gesture Camera Developer Kit*. Retrieved April 21, 2013, from http://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk

Dourish, P. (2001). *Where the Action is: the fondation of embodied interaction.* Cambridge, MA, USA: MIT Press.

*FoldIt*. (2013, April 30). Retrieved from http://fold.it/portal/

Gillian, N., Knapp, R., & O'Modhrain, S. (2011). An Adaptive Classification Algorithm For Semiotic Musical Gestures. *8th International Conference on Sound and Music Computing.* Padova, Italy.

Hollan, J., & Stornetta, S. (1992). Beyond Being There. *CHI '92 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 119-125). Monterey, California, USA: ACM.

Ishii, H., Kobayashi, M., & Arita, K. (1994). Iterative design of seamless collaboration media. *Communications of the ACM , 37* (8), 83-97.

Johansen, R. (1988). *GroupWare: Computer Support for Business Teams.* New York, NY, USA: The Free Press.

Kooima, R. (2009). *Generalized Perspective Projection.* Technical, Louisiana State University, Center for Computation & Technology.

*Leap Motion*. Retrieved April 21, 2013, from http://leapmotion.com

*Libjpeg-turbo*. Retrieved April 21, 2013, from http://sourceforge.net/projects/libjpeg-turbo

Maes, P., Darrel, T., Blumberg, B., & Pentland, A. (1997). The ALIVE system: wireless, full-body interaction with autonomous agents. *Multimedia Systems - Special issue on multimedia and multisensory virtual worlds , 5* (2), 105-112.

Maimone, A., Bidwell, J., Peng, K., & Fuchs, H. (2012). Enhanced personal autostereoscopic telepresence system using commodity depth cameras. *Computers & Graphics , 36* (7), 791-807.

*Microsoft Kinect*. Retrieved April 20, 2013, from http://xbox.com/en-US/kinect

Morikawa, O., & Maesako, T. (1998). HyperMirror: toward pleasant-to-use video mediated communication system. *Proceedings of the conference on Computer supported cooperative work* (pp. 149-158). Seattle, Washington, USA: ACM.

Morikawa, O., Hashimoro, R., & Yamashita, J. (2003). Self reflection can substitute eye contact. *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (pp. 944-945). Ft. Lauderdale, Florida, USA: ACM.

Nescher, T., & Kunz, A. (2011). An interactive whiteboard for immersive telecollaboration. *The Visual Computer: International Journal of Computer Graphics - Special Issue on CYBERWORLDS 2010 , 27* (4), 311-320.

*NiTE*. Retrieved Mai 9, 2013, from http://www.primesense.com/solutions/nite-middleware/

*Oblong g-speak/Mezzanine*. Retrieved April 20, 2013, from http://oblong.com

Oikonomidis, I., Kyriazis, N., & Argyros, A. (2011). Efficient model-based 3D tracking of hand articulations using Kinect. *Proceedings of the British Machine Vision Conference* (pp. 101.1-101.11). Dundee, Scotland: BMVA Press.

*OpenNI*. Retrieved April 21, 2013, from http://openni.org

*Polycom TelePresence*. Retrieved April 20, 2013, from http://polycom.com/telepresence

R Core Team. (2012). R: A language and environment for statistical computing. *http://R-project.org* . Vienna, Austria: R foundation for Statistical Computing.

Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., & Fuchs , H. (1998). The office of the future: a unified approach to image-based modeling and spatially immersive displays. *SIGGRAPH '98: Conference on Computer graphics and interactive techniques* (pp. 179-188). New York, NY, USA: ACM.

*Skype*. Retrieved April 20, 2013, from http://skype.com/intl/en-us/home

Slater, M., Sadagic, A., Usoh, M., & Schroeder, R. (2000). Small-Group Behavior in a Virtual and Real Environment: A Comparative Study. *Presence: Teleoperators and Virtual Environments , 9* (1), 37-51.

Stein, M., Xiao, X., Tompson, J., Hendee, C., Perlin, K., & Ishii, H. (2012). ARCADE: A System for Augmenting Gesture-Based Computer Graphic Presentations. *Proceedings of SIGGRAPH Computer Animation Festival* (p. 77). Loa Angeles, CA, USA: ACM.

Tang, A., Neustaedter, C., & Greenberg, S. (2007). VideoArms: Embodiments for Mixed Presence Groupware. In *People and Computers XX — Engage* (pp. 85-102). London: Springer.

Tanner, P., & Shah, V. (2010). Improving remote collaboration through side-by-side telepresence. *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (pp. 3493-3498). Atlanta, Georgia, USA: ACM.

von Kapri, A., Rick, T., & Feiner, S. (2011). Comparing steering-based travel techniques for search tasks in a CAVE. *VR '11: Virtual Reality Conference* (pp. 91-94). IEEE.

W.M. Keck Center for Active Visualization in the Earth Sciences. *KeckCAVES.* Retrieved April 20, 2013, from http://keckcaves.org

Wang, R., & Popovic, J. (2009). Real-time hand-tracking with a color glove. *Transactions on Graphics , 28* (3), 63:1-63:8.

Wang, R., Paris, S., & Popovic, J. (2011). 6D hands: markerless hand-tracking for computer aided design. *Symposium on User interface software and technology* (pp. 549-558). Santa Barbara, California, USA: ACM.

Wilson, A. D. (2006). Robust computer vision-based detection of pinching for one and two-handed gesture input. *Symposium on User interface software and technology* (pp. 255-258). Montreux, Switzerland: ACM.

Wilson, A., Izadi, S., Hilliges, O., Garcia-Mendoza, A., & Kirk, D. (2008). Bringing physics to the surface. *UIST '08: Symposium on User interface software and technology* (pp. 67-76). Monterey, CA, USA: ACM.

Wolff, R., Roberts, D. J., Steed, A., & Otto, O. (2007). A review of telecollaboration technologies with respect to closely coupled collaboration. *International Journal of Computer Applications in Technology , 29* (1), 11-26.

*zSpace by Infinite Z.* Retrieved April 21, 2013, from http://zspace.com

# ACRONYMS

| | |
|---|---|
| 3D | three-dimensional |
| ANBC | adaptive naïve bayes classifier |
| ANOVA | analysis of variance |
| CAD | computer-aided design |
| CSCW | computer-supported cooperative work |
| DOF | degrees of freedom |
| DVR | Desktop VR |
| F2F | Face-to-Face |
| FPS | frames per second |
| GPU | graphics processing unit |
| HCI | human computer interaction |
| OSC | Open Sound Control |
| SbS | Side-by-Side |
| UDP | user datagram protocol |
| VR | Virtual Reality |

# APPENDIX A: QUESTIONNAIRE

Questions

Age:

Gender: Male Female

Kinect/Wii Experience (daily, twice a week, once a week, once a month, once a year, never):

---

Please order the interfaces in terms of co-presence from most favorite to least favorite.

_____ side-by-side

_____ face-to-face hands

_____ face-to-face skeleton

How would the ordering look like if you would rate them regarding game play?

_____ side-by-side

_____ face-to-face hands

_____ face-to-face skeleton

Which interface was your favorite? Why?

***Manipulation:***

On a scale from 1-5 (1= least frustrating, 5= most frustrating) how frustrating did you experience the 3D manipulation?

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

How intuitive do you think the 3D manipulation is (1-least intuitive, 5 most intuitive)?

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

What did you want to do and could not?

Other suggestions?

***Side-by-side***

Co-presence:

1. In the remote meeting to what extent did you have the sense of being together with the other person?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

2. Continue to think back about the remote meeting. To what extent can you imagine yourself being now with the other person in that room?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

3. Please rate how closely your sense of being together with the other person in a real-world setting resembles your sense of being with them in the virtual room.

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

Suggestions?

### *Face-to-face Hands*

Co-presence:

1. In the remote meeting to what extent did you have the sense of being together with the other person?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

2. Continue to think back about the remote meeting. To what extent can you imagine yourself being now with the other person in that room?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

3. Please rate how closely your sense of being together with the other person in a real-world setting resembles your sense of being with them in the virtual room.

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

Suggestions?

### ***Face-to-Face skeleton***

Co-presence:

1. In the remote meeting to what extent did you have the sense of being together with the other person?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

2. Continue to think back about the remote meeting. To what extent can you imagine yourself being now with the other person in that room?

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

3. Please rate how closely your sense of being together with the other person in a real-world setting resembles your sense of being with them in the virtual room.

not at all completely

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

Suggestions?

# APPENDIX B: CODE SAMPLES

## *B.1 Hand pose detection*

```
    /// member variables for hand pose calculation in hand.h
    /// for calculating the hand mask
    IplImage *src, *dst, *res, *roi;

    /// for contour calculations on hand mask
    CvSeq               *contour;
    CvMemStorage        *contour_storage;
    CvMemStorage        *convexity_storage;

CvBox2D hand::get_min_ellipse(CvSeq *contour)
{
    CvBox2D ellipse;
    CvSeq * outer_contour = contour;

    double area = 0.0;
    double tmp_area;

    //only interested in outer contours here
    for( ; outer_contour != 0; outer_contour = outer_contour->h_next )
    {
        tmp_area = cvContourArea( outer_contour);
        if(tmp_area > area)
        {
            area = tmp_area;

            //cvFitEllipse2 only works with at least 5 points
            //limit the size of the contour to 100 points
            int num_points = outer_contour->total;
            if((num_points >= 5) && (num_points <= 100))
            {
                ellipse = cvFitEllipse2(outer_contour);
            }
        }
    }
    return ellipse;
}

///calculate hand mask dependent on current hand position
void hand::handMask(ofxDepthGenerator * depthGenerator){
    // use native OpenCV to get mask of hand dependent on depth pixels,
    // user pixels and area around hand
    cvSetData(src, depthGenerator->getDepthPixels(this-> projectedTrackedHand->z - depth,
this-> projectedTrackedHand->z + depth), width);
    cvSetData(dst, userMask->getPixels(), width);
    cvZero(roi);
    cvZero(res);

    if(projectedTrackedHand->z == 0){
        projectedTrackedHand->z = 1;
    }
    float adjustedRadius = radius*900 / (projectedTrackedHand->z);

    //define roi as circle around hand
    cvCircle(roi,
             cvPoint(projectedTrackedHand->x, projectedTrackedHand->y),
             adjustedRadius,
             CV_RGB(255, 255, 255),
             -1, 8, 0);

    // src & dst if roi==1
    cvAnd(src, dst, res, roi);
}
```

```
///calculate contour on res
void hand::calcContours(IplImage *img) {
    //use openCV contour
    contour_storage = cvCreateMemStorage(0);
    contour = 0;
    IplImage* ipltemp = cvCloneImage(img);

    // CV_RETR_CCOMP: retrieves all of the contours and
    // organizes them into a two-level hierarchy
    cvFindContours(ipltemp, contour_storage, &contour, sizeof(CvContour),
                   CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE );

    cvReleaseImage(&ipltemp);
}

// Check if hand is grabbing
bool hand::Grabbing(CvSeq *ctr, CvMemStorage *ctr_storage){

    CvSeq* largest_contour = 0;
    CvSeq* tmp_contour = ctr; // to not loose pointer on contour
    int largest_area = 0;
    int tmp_area;

    for( ; tmp_contour != 0; tmp_contour = tmp_contour->h_next )
    {
        tmp_area = cvContourArea(tmp_contour);
        if(tmp_area>largest_area)
        {
            largest_area = tmp_area;
            largest_contour = tmp_contour;
        }
    }

    if(!largest_area) return -1.0f;

    convexity_storage = cvCreateMemStorage(0);

    CvSeq* hull = cvConvexHull2( largest_contour, 0, CV_CLOCKWISE, 0 );

    CvSeq* defect = cvConvexityDefects( largest_contour, hull, convexity_storage );

    largest_depth =0.0f;
    float tmp_depth;

    CvConvexityDefect *defectArray = 0;
    defectArray = (CvConvexityDefect*)malloc(sizeof(CvConvexityDefect)*defect->total);
    cvCvtSeqToArray(defect, defectArray, CV_WHOLE_SEQ);

    for(int i = 0; i<defect->total; i++){
        tmp_depth = defectArray[i].depth;
        if(largest_depth < tmp_depth) largest_depth = tmp_depth;
    }

    if( convexity_storage != NULL ) { cvReleaseMemStorage( &convexity_storage ); }

    // It seems that the value 15.0 is for hand area around 5000, so:
    float min = 14.0 * sqrt(largest_area/6000.0);

    if(largest_depth < min) return true;
    else return false;
}

///after checking for pinching etc free contour storage again
void hand::freeContours(){
    if( contour_storage != NULL ) { cvReleaseMemStorage(&contour_storage); }
}
```

```
void hand::updateLocal(ofxDepthGenerator * depthGen){
    active = true;
    //get hand mask
    handMask(depthGen);

    // calc contour on res
    calcContours(res);

    // check grabbing
    bool grab  = Grabbing(contour, contour_storage);

    // use classifier to predict hand pose
    if (USE_CLASSIFIER){
        double bestLoglikelihood = 0;
        GRT::Vector<double> logLikelihood;
        GRT::Vector<double> testSample(NUM_VAR);

        CvBox2D ellipse = get_min_ellipse(contour);
        float ratio;
        if (ellipse.size.height != 0){
            ratio = ellipse.size.height/ellipse.size.width;
            if (ratio < 1){
                ratio = 1/ratio;
            }
        } else {
            ratio = 0;
        }
        ratio = ratio*1000;
        double normalizedDefect = largest_depth*realTrackedHand->z;

        testSample[0] = realTrackedHand->z; // var1: depth of hand position
        testSample[1] = normalizedDefect; // var2 : largest defect
        testSample[2] = ratio; // var3

        if (anbc->predict(testSample,bestLoglikelihood,logLikelihood) == 2){
            grab = true;
        } else {
            grab = false;
        }
    }

    timer += ofGetElapsedTimeMillis()-previousElapsedTime;
    previousElapsedTime = ofGetElapsedTimeMillis();

    handAction previousAction = action;
    if (grab) {
        action = GRAB;
    } else {
        action = NO_ACTION;
    }

    // STATE RESISTANCE:
    if (previousAction != action){
        if (timer < STATE_TIMEOUT_MILIS){ // noise:
            action = previousAction;
        }
    } else {
        timer = 0; // reset timer: still in state
    }

    freeContours();
}
```

### *B.2 Setting up the adaptive naïve Bayes classifier*

```
// SETTING UP THE ANBC: ======================
    gamma = 2;
    model_filepath = "handActionModel.txt";
    numVar = 3;
    labelledTrainingData = LabelledTrainingData(numVar);
// ===========================================


// TRAINING THE ANBC
if(isTrainingGrabbing || isTrainingNoAction || isTrainingSideways){
        if(timer < 10){
                timer += ofGetElapsedTimef() - previousTime;
                previousTime = ofGetElapsedTimef();

                // code for gathering data: ------------------------------------
                UINT classLable;
                if(isTrainingGrabbing){classLable = GRAB;};
                if(isTrainingNoAction){classLable = NO_ACTION;};
                if(isTrainingSideways){classLable = SIDEWAYS;};

                Vector<double> trainingSample(numVar);

                trainingSample[0] = right.Z; // var1: depth of hand position
                trainingSample[1] = normalizedDefect; // largest defect
                trainingSample[2] = ratio; // var3
                //trainingSample[3] = normalizedDefect;

                labelledTrainingData.addSample(classLable, trainingSample);
                // -------------------------------------------------------------
        } else {
                if (isTrainingGrabbing){
                        std::cout << "GRAB TRAINING IS OVER!\n";
                        isTrainingGrabbing = false;
                }
                if (isTrainingNoAction){
                        std::cout << "NO ACTION TRAINING IS OVER!\n";
                        isTrainingNoAction = false;
                }
                if (isTrainingSideways){
                        std::cout << "SIDEWAYS TRAINING IS OVER!\n";
                        isTrainingSideways = false;
                }
        }
}
```

```
void testApp::saveTrainingData(bool isExit){
    // SAVE THE TRAINING DATA: ======================
    UINT errorID = 0;
       if( !anbc.train( labelledTrainingData, gamma, errorID) ){
        cout << "ERROR: Failed training the ANBC algorithm\n";
       }

    time_t     now = time(0);
    struct tm  tstruct;
    char       buf[80];
    tstruct = *localtime(&now);
    strftime(buf, sizeof(buf), "%Y_%m_%d_%X", &tstruct);

    stringstream filepath;
    if (isExit){
        filepath << "HandAction_"<< buf <<"_EXIT.txt";
    } else {
        filepath << "HandAction_"<< buf <<".txt";
    }
    model_filepath = filepath.str();
    cout << "Saving model to: " << model_filepath << endl<< "....." << endl;
       if( anbc.saveANBCModelsToFile(model_filepath) ){
        cout << "ANBC model saved to file\n";
       }else{
        cout << "ERROR: Failed to save ANBC model to file\n";
       }
    // ==============================================
}


if(useTrainingData){
       // Use the training data: ------------------------------------------------
        double bestLoglikelihood = 0;
         Vector<double> logLikelihood;
       Vector<double> testSample(numVar);

       testSample[0] = right.Z; // var1: depth of hand position
       testSample[1] = normalizedDefect;
       testSample[2] = ratio; // var3
       //testSample[3] = normalizedDefect;

       action = (handAction) anbc.predict(testSample,bestLoglikelihood,logLikelihood);
       //-------------------------------------------------------------------------
}
```