THE UNIVERSITY OF CALGARY

Embodiments in Mixed Presence Groupware

By

Anthony Hoi Tin Tang

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

JANUARY 2005

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommended to the Faculty of Graduate Studies for acceptance, a thesis entitled “Embodiments in Mixed Presence Groupware” submitted by Anthony Tang in partial fulfillment of the requirements for the degree of Master of Science.

____

Supervisor, Saul Greenberg
Department of Computer Science

____

Theresa Kline
Department of Psychology

____

Ehud Sharlin
Department of Computer Science

____

Date

Abstract

In this thesis, I define and explore Mixed Presence Groupware (MPG): software that connects distributed groups of collaborators together, allowing collocated individuals to work together on a shared display while simultaneously working with other, remote groups in the same digital workspace. In my explorations of this new class of groupware, I articulate a problem unique to MPG workspaces called presence disparity, where collaborators focus their collaborative energies toward collocated collaborators while ignoring their remote counterparts. I propose that the root cause of this problem is the poor representational properties of embodiments for remote collaborators, and develop a theory about embodiments for MPG workspaces. I present a video-based embodiment technique called VideoArms that addresses the presence disparity problem by following the design guidelines set out by the theory. Finally, I evaluate this embodiment technique, demonstrating and critiquing its effectiveness in mitigating presence disparity.

Publications

Materials, ideas, and figures from this thesis have appeared previously in the following publications:

Tang, A., Boyle, M. and Greenberg, S. (in press). **Display and Presence Disparity in Mixed Presence Groupware**. To appear in *Journal of Research and Practice in Information Technology*.

Tang, A., Neustaedter, C. and Greenberg, S. (2004). **Embodiments and VideoArms in Mixed Presence Groupware**. Report 2004-741-06, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4, March.

Tang, A., Neustaedter, C. and Greenberg, S. (2004). **Embodiments for Mixed Presence Groupware**. Report 2004-769-34, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4, March.

Tang, A., Neustaedter, C. and Greenberg, S. (2004). **VideoArms: Supporting Remote Embodiment in Groupware**. In *Video Proceedings of the ACM CSCW Conference on Computer Supported Cooperative Work*. (November 6-10, Chicago, Illinois). ACM Press.

Acknowledgements

It is funny putting my name as the sole author of this thesis. I owe an incredible debt to many individuals for their inspiration and expertise—I only hope I can help so much in the future. This thesis is really the culmination of the efforts of a lot of people, and I am glad I have this space to say so.

To my supervisor Saul, you have been a wonderful source of inspiration, odd humor, and insight. Beyond being a supervisor, you have been a friend and a true motivator, helping me to find my inner strength and passion.

To my best friend Cheryl, thanks for all the long nights you stayed up with me, supporting me and helping me in every way that you did. You have a way of always making me laugh, especially when I need it most. I have grown as a human being because you have been in my life.

To Carman, thanks for being there every single time I needed you. You have been a wonderful friend, a great foosball partner, a source of great laughs, a mentor, a role-model and a hero for me. If I grow up to be half the man you are, I will be the most surprised.

To Russell, you have been a steady, positive influence on my life. Because of you, I have become a better researcher, brother, friend, cook, and foosball player.

To my friends at the Interactions Lab, particularly Ed, Eric, Gregor, Kathryn, Nelson, Mike, Petra, Sheelagh, and Stacey, thank you for all the good times, the inspiration, and the fun. I enjoyed every single minute of it.

Lastly, to my brother Jonathan, you personify effort and passion. When I am down the most, I think of you, the first and the last player to dive for every ball on the volleyball court. It fills me with pride to be able to say that you are my brother—the best anyone could ask for.

Dedication

For my parents, Eva and Tom.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

In this thesis, I define and explore *mixed presence groupware (MPG)*: software that connects distributed groups of collaborators together allowing collocated individuals to work together on a shared display while simultaneously working with other, remote groups in the same digital workspace. To this point, the study of real-time groupware can be divided into two separate groups (Baecker, Grudin, Buxton & Greenberg, 1995): *distributed groupware*, software connecting physically distributed individuals, and *collocated groupware*, software allowing collocated individuals to work together. MPG attempts to transcend the boundaries of physical separation by providing the technical means for individuals to work with both collocated and distributed collaborators simultaneously.

Yet the unique arrangement of collaborators supported by MPG also sets up its main problem: *presence disparity*. In this thesis, I articulate this unique problem, where collaborators focus their energies on collocated collaborators at the expense of their distributed counterparts. This imbalance occurs because individuals have an easier time working with people who are physically present compared to those who are not. Presence disparity is a problem because it negatively impacts the collaborative efforts of small groups.

I propose that a root cause of presence disparity is that remote participants are poorly represented in the workspace by virtual embodiments. In the real world, a person's embodiment is one's physical body: how one is seen by others, what one does, and how one affects the world is always done through the physical body. In a digital workspace, one's embodiment is somehow "coded" into the system through some kind of graphical surrogate. For example, a very common graphical embodiment in a collaborative workspace is the mouse cursor (Greenberg, Gutwin & Roseman, 1996). Cursors, like physical bodies, inform others of what we are doing and where we are located. This information is important since our actions in a mutually shared workspace affect the work of others. One's embodiment is the *primary mechanism* through which others maintain an awareness of one's activities in the shared workspace.

In this thesis, I will establish that an effective embodiment accurately reflects the body it represents. I will then show that better embodiments aid collaboration in MPG by addressing the presence disparity problem, in particular, by reducing the difficulty of working with remote collaborators. I present an embodiment system called VideoArms as a concrete instance of the design features that I believe should make up an embodiment for in MPG systems. In addition to the system itself, I present a preliminary evaluation of VideoArms that demonstrates and critiques its effectiveness in supporting collaboration in MPG scenarios.

1.1 Problem scope

The scope of this thesis (Figure 1.1) is confined to a concentrated sub-area of human-computer interaction. Starting at the outer shell of Figure 1.1, the study of human-computer interaction looks generally at interactive interfaces and processes that occur between technology and its users. Computer Supported Cooperative Work (CSCW) is a discipline within human-computer interaction that seeks to positively augment collaborative work with technology. Some research in CSCW focuses on supporting real-time interaction (such as that provided by a telephone) between collaborators. One medium for supporting real-time collaborative work is a shared visual workspace. A shared visual workspace is best described as the digital analogue of whiteboards or tables. The way in which users interact with these digital, shared visual workspaces is typically divided into two categories: distributed groupware, and collocated groupware. *Distributed groupware* supports collaborators working at a distance, or in physically disparate locations. In contrast, *collocated groupware* supports collaborators who are working together in the same physical space.

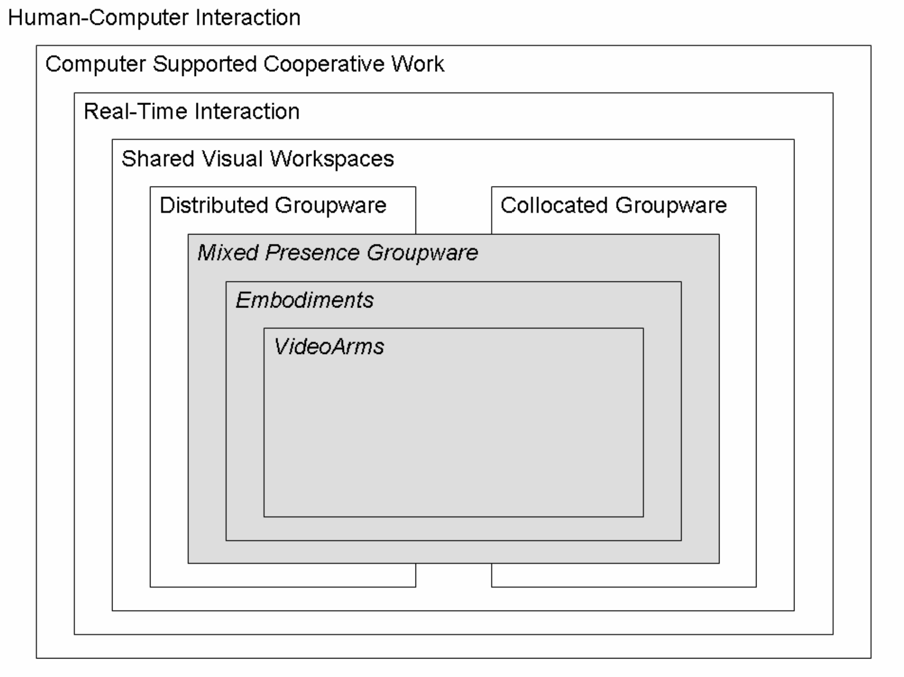
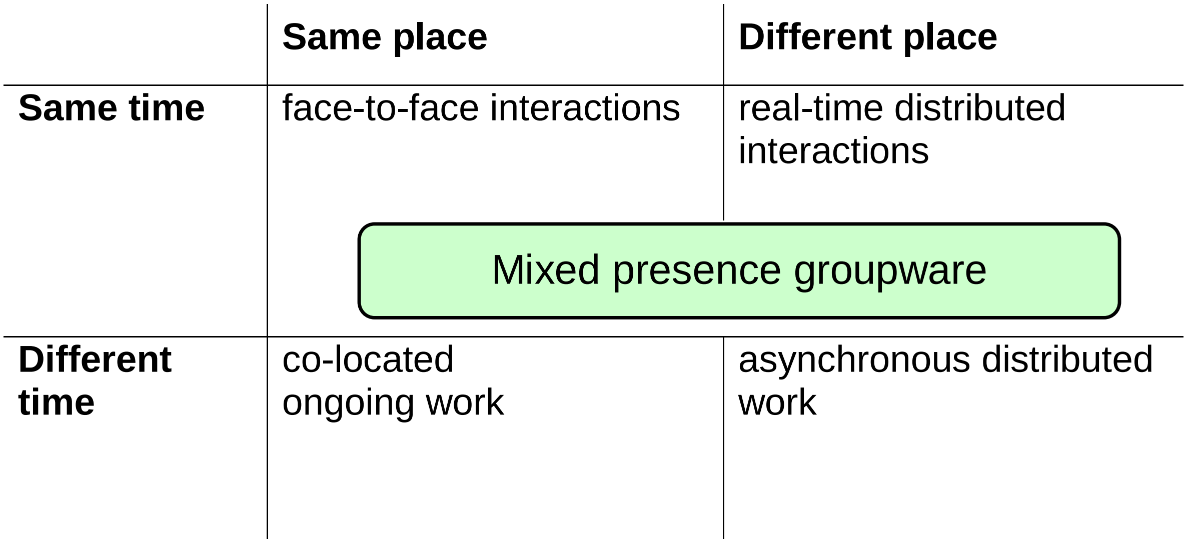


Figure 1.1. Scope of this thesis. The inner, shaded rings represent the work presented in this thesis.

I define *mixed presence groupware* as a combination of distributed and collocated groupware, supporting collaborators who may be collocated, and some who are physically distributed. Within the context of MPG, this thesis examines the role of *embodiments*, which are representations of collaborators that provide collaborators with an awareness of what others are doing in the workspace. Finally, the thesis presents VideoArms, a specific type of embodiment intended for use with MPG.

Three concepts from the above discussion are extremely important in this thesis: *mixed presence groupware*, *shared visual workspaces*, and *embodiments*. These concepts deserve further treatment, which I give below. I defer the discussion of embodiments to Chapter 3 as their importance is made clearer in Chapter 2.



	Same place	Different place
Same time	face-to-face interactions	real-time distributed interactions
Different time	co-located ongoing work	asynchronous distributed work

Figure 1.2. Mixed presence groupware in the space – time groupware matrix.

1.2 Mixed presence groupware

The time/space taxonomy of groupware (Figure 1.2) categorises applications based on where and when collaborators use them (Baecker et al, 1995). This taxonomy partitions groupware into four quadrants based on style of use and work practices:

- *same time–same place* systems supporting face-to-face interactions (e.g. Bier & Freeman's (1991) Multi-Device, Multi-User, Multi-Editor),
- *same time–different place* systems supporting real time distributed interactions (e.g. instant messenger applications),
- *different time–different place* systems supporting asynchronous work (e.g. email), and
- *different time–same place* systems supporting co-located, on-going tasks (e.g. applications supporting shift work).

Many applications have been designed to fit within a quadrant. For example, MMM (Multi-Device, Multi-User, Multi-Editor) cleanly fits within the same time-same place cell because it supports co-located collaborators sharing a single display using multiple mice (Bier & Freeman, 1991). Systems in this cell are typically called *collocated groupware systems*. Since most collocated groupware systems are built with a single shared display, the term *single display groupware (SDG)* is used interchangeably with collocated groupware even though SDG is properly a subset of the former. In contrast, the popular instant messaging system MSN Messenger is an example of a system supporting real-time, distributed collaboration: individuals are physically separated, but the communications medium supports instantaneous interaction. MSN Messenger fits into the same time-different place cell, and these systems are typically called *distributed groupware*.

However, this quadrant view of groupware is limiting (Baecker, 1992); in practice, collaborative practices cross the boundaries laid out by the taxonomy. For instance, collaborators working two time zones apart, but maintaining 9:00am-5:00pm work hours will work together for six hours, but alone for a total of four hours. TeamWave Workplace's "rooms" metaphor supports this transcendence of the "time" boundary (Greenberg & Roseman, 2003): as multiple people enter a virtual room, they can interact synchronously over all items within a room. Yet, they can also leave items in a room for absent people to work on later, thus permitting asynchronous interaction.

Likewise, mixed presence groupware (MPG) supports real-time work by both co-located and distributed collaborators, thereby spanning the same place-different place quadrants at the top of Figure 1.2. Figure 1.3 gives an illustrative example, where several distributed groups of co-located people work over various physical displays containing a common shared visual workspace. As seen in the figure, the physical display may be a horizontal tabletop display, a vertical presentation display, or even a conventional monitor. All participants have their own input devices, and all can interact with the workspace at the same time with their actions being immediately reflected on all displays. The lower right image in Figure 1.3 illustrates this concept in a virtual space.

MPG's relationship with the top two cells of the taxonomy is important. The intellectual foundation of MPG owes part of its parentage to collocated groupware research, because each group of collaborators in Figure 1.2 is working in a real-time, collocated

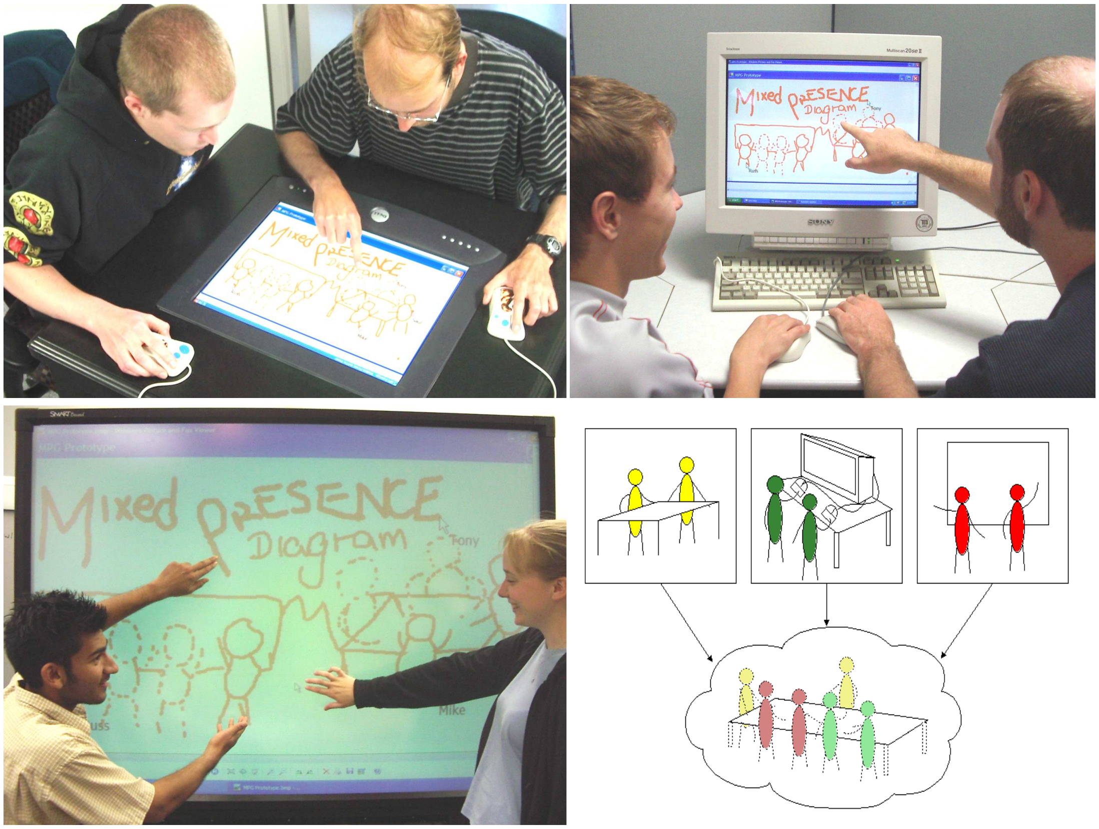


Figure 1.3. Three teams working in a conceptual MPG setting over three connected displays, stylized as a virtual table in the bottom right.

situation. Secondly, since groups are working with other physically distributed groups, MPG also draws on research in distributed groupware.

1.2.1 Existing mixed presence groupware systems

Surprising, very few examples of mixed presence groupware exist in the literature, let alone mixed presence groupware supporting shared visual workspaces. Perhaps the most common examples of MPG are based on video conferencing technology. These systems typically use a video channel that transmits the image of participants' work over a drawing surface, or special overlays are used allowing people to annotate over this video image. Some research systems even provide a shareable video-based drawing area by overlaying the images of multiple video cameras (Tang & Minneman, 1991a; Tang & Minneman, 1991b; Ishii & Kobayashi, 1992). While demonstrations show these systems as a means for connecting two distributed collaborators, typically, they are not technically limited to only two collaborators. An unfortunate constraint of these systems is that participants cannot alter artefacts on the drawing surface introduced by remote participants.

In the non-video world, we see games often implemented as a type of MPG. A commercial example of MPG is Halo, a multi-player game for Microsoft's Xbox. Co-located players can interact through a split-screen, and distributed groups of players can be connected together by connecting several Xboxes together. All players and their actions are visible in each person's scene.

Finally, people often work in an MPG-like mode even when the software does not support it. For instance, instant messengers explicitly support only one user per terminal chatting with others on their own terminals. However, others may chat "over the shoulder" by telling the co-located partner what to type, or by taking control of the mouse and keyboard.

My focus on MPG is distinct from this prior work. First, I am interested in supporting scenarios that allow multiple co-located teams to gain equal access to a single shared drawing surface. Second, all participants have their own input device, where each participant can manipulate the shared space—even simultaneously—at any time (Figure 1.3).

1.2.2 Shared visual workspaces

In this thesis, I am only interested in groupware that supports shared visual workspaces. A shared visual workspace is one where participants can create, see, share and manipulate artefacts within a bounded space. Real world examples include whiteboards and tabletops, which allow groups to collaborate using tools like markers, paper, tape, and scissors. Electronic counterparts to shared workspaces have been developed as distributed groupware, single display groupware, and to a much lesser extent, mixed presence

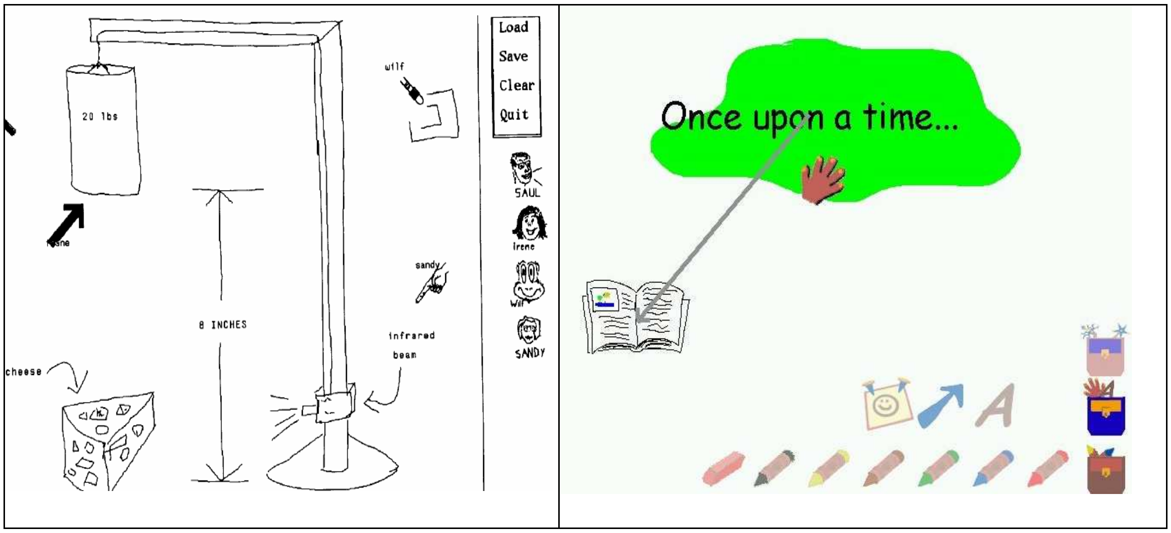


Figure 1.4. Screen captures from GroupSketch (left) and KidPad (right). The image of GroupSketch shows a mousetrap brainstorming session with an iconic participant on the right to show all remote collaborators in the workspace (Greenberg & Bohnet, 1991). The image of Kidpad shows shared tools along the bottom and right of the workspace while a participant list is unnecessary since all participants work side-by-side (Druin et al., 1997).

groupware. These environments also provide digital counterparts to the physical tools of the real world visual workspaces.

Much of the research in CSCW has focused on shared visual workspaces because of their simplicity and inherent flexibility. To give a flavour of such digital shared visual workspaces, I will describe two systems from the CSCW literature that implement a shared visual workspace: GroupSketch, a distributed groupware system, and Kidpad, a collocated groupware system. While many other similar systems exist, these two are classic exemplars in their respective areas.

GroupSketch (Greenberg & Bohnet, 1991) was a computer-based system supporting multiple distributed users in a shared visual workspace (Figure 1.4, left). These users could be geographically distributed so long as their computers were connected via a network connection. Each user in the communal work surface was represented as a labeled mouse pointer. Users could simultaneously list, draw and gesture on the work surface, and their actions would be immediately reflected on collaborators' screens.

Kidpad (Druin, Stewart, Bederson & Hollan, 1997) is a similar system, supporting simultaneous drawing activities for multiple collaborators working in a shared visual workspace, except for one crucial difference: instead of supporting multiple distributed collaborators, Kidpad provides support for multiple users on the same computer (Figure 1.4, right). Kidpad is an early example of a system that provided simultaneous input device support for a single computer.

1.3 Problem statement

The scope of mixed presence groupware research extends far beyond this thesis. I confine the problem space of this work in several important ways. First, I am interested in real-time collaboration: collaboration that occurs in up-to-the-minute scenarios, such as in design meetings or brainstorming sessions. This work does not concern collaboration that necessarily occurs before and after such sessions, or asynchronous interactions. Secondly, this thesis focuses on small distributed teams, such as two groups of two individuals per group. Finally, I am primarily interested in collaboration that occurs in only shared visual workspaces: environments that support creating, moving, removing and annotating workspace objects in a visual manner.

In this research, I address three problems:

1. **We do not understand the technical and social challenges inherent in mixed presence groupware systems.** Because no documented MPG systems exist, I need to architect MPG systems myself. Although I can draw on existing literature to understand the dynamics of collaboration, I need to observe how the unique physical arrangement of users in an MPG system affects collaborative dynamics.

2. **We do not fully understand the role of embodiments in the specific context of mixed presence groupware.** Given my observations of MPG systems in use and existing literature on collaboration in shared visual workspaces, I need to understand how people use the visible aspects of their collaborators to facilitate collaboration.

3. **We do not know what embodiment techniques are appropriate for mixed presence groupware.** Because of Problem 2, we cannot yet build effect effective

embodiments for mixed presence groupware. Indeed, what makes up an "effective" embodiment in mixed presence groupware is unknown.

1.4 Goals

My research as presented in this thesis addresses the three problems above. I have three primary goals in this research:

1. **I will lay out the foundations of Mixed Presence Groupware systems and describe my preliminary observations of their use**. Prior work has mainly considered distributed and co-located groupware in separate contexts. Mixed presence groupware considers both these contexts simultaneously, introducing a broader view of how groupware and shared computerized workspaces can be used in the workplace. I will build MPG systems, and describe an architectural pattern for future research in MPG. From my experiences in building MPG, and my observations of MPG's use, I will articulate two new problems particular to MPG: presence disparity, and display disparity.

2. **I will develop a set of design requirements for embodiments in mixed presence groupware systems.** Based on observations of MPG and prior work investigating the importance of physical bodies in individual and collaborative work, I will articulate a set of design requirements for embodiments that highlight the importance of supporting feedthrough, consequential communication and gestures. To show the value of these requirements, I will analyse existing embodiment techniques for distributed groupware supporting group work in a real-time shared visual workspace.

3. **I will implement and evaluate a novel video-based embodiment technique**. I will develop a video-based embodiment technique that naturally allows users to use a wider range of gestures than existing embodiment techniques like mouse cursors. Through an observational evaluation, I will demonstrate that this embodiment technique supports the design criteria set out in Goal 2.

1.5 Overview

In Chapter 2, I present an architectural pattern for MPG, describing an initial prototype called MPGSketch as a case study. I then describe how the system produced surprisingly unusual collaborative dynamics in practice. These problems are articulated and defined as presence and display disparity. (Problem 1, Goal 1)

Chapter 3 takes a step back where I lay out the necessary background knowledge for groupware embodiments, and synthesise this information within the context of mixed presence groupware. I distill knowledge from the literature about the role embodiments play in maintaining workspace awareness, showing how existing groupware embodiments do not completely fulfill the role a physical body plays in face-to-face settings. I conclude the chapter by deriving four principles that MPG embodiments and MPG systems should fulfill to mitigate presence disparity. (Problem 2, Goal 2)

In Chapter 4, I put these principles to practice by building a proof of concept system called VideoArms. I describe how the prototype was built, the algorithms I chose to use, and alternative methods of building a similar system. (First half of Problem 3, First half of Goal 3)

Chapter 5 documents a preliminary observational study I ran to observe VideoArms in use. I describe my observations and show how the results of the study validate the principles I lay out in Chapter 3. I also use Chapter 5 to put the results of the study in context and to understand the implications it has for mixed presence groupware research. (Second half of Goal 3, second half of Problem 3)

Finally, Chapter 6 concludes by placing this research and the concept of mixed presence groupware into the broader context of computer supported cooperative work. I discuss the contributions of this work, and lay out the path for future work.

Chapter 2. Explorations into mixed presence groupware

Mixed presence groupware supports simultaneous collaboration with both collocated and remote collaborators. Unfortunately, because we have yet to see MPG applications, we do not yet understand the technical and social challenges inherent in their design (Problem 1). In this chapter, I describe the design and implementation of a prototype application called MPGSketch, which served as a means to gain this initial understanding of MPG. From the technical perspective, I describe a generalizable architectural pattern for MPG system construction. From the social perspective, I identify and describe two problems revealed through the iterative process of designing and testing the MPGSketch prototype: the *display disparity problem*, and the *presence disparity problem* (Goal 1). The latter problem, unique to MPG workspaces, motivates the remainder of my thesis.

2.1 MPGSketch: A mixed presence groupware drawing system

In this section, I describe the functionality of MPGSketch, and its implementation. I follow with a fairly technical presentation of an architectural pattern for prototyping MPG applications.

2.1.1 Description

I began my investigation by implementing and using MPGSketch, a simple MPG real-time shared drawing application. MPGSketch allows participants to sketch together over an empty surface, or over an image taken from a file, a captured webcam snapshot, or a

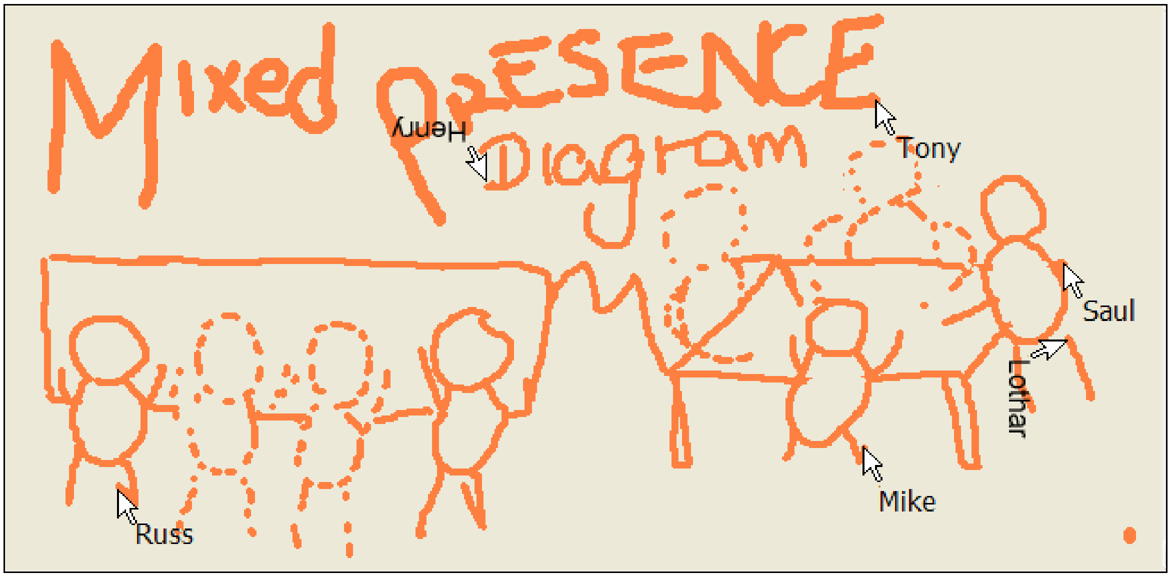


Figure 2.1. MPGSketch with six participants, each with a telepointer that reflects his or her position in the workspace.

captured screen of the desktop. A screen capture of MPGSketch is shown in Figure 2.1, while Figure 1.3 shows a simulated image of participants sketching over the workspace.

Each person has his or her own pointing device for input (e.g. finger on touch-sensitive table, pen on vertical whiteboard, or mouse). Multiple cursors, labelled with their owners' names, show the location and movement of the pointing devices in the workspace. The shared workspace presents all participants with the drawing as it evolves over time, as well as the position of all the cursors within it. Participants can draw on the display simultaneously and at any time, and the drawing actions are reflected immediately on all displays. What makes MPGSketch an MPG application is that, as illustrated in Figure 1.3, several individuals can work on a single display, and that this display is connected to remote displays being worked on by others (a stylized representation appears in Figure 2.2).

2.1.2 Implementation of MPGSketch

At a high level, the functional requirements of an MPG system are fairly straightforward. However, because MPG systems are extremely rare, I detail my approach to building MPG systems here (Problem 1).

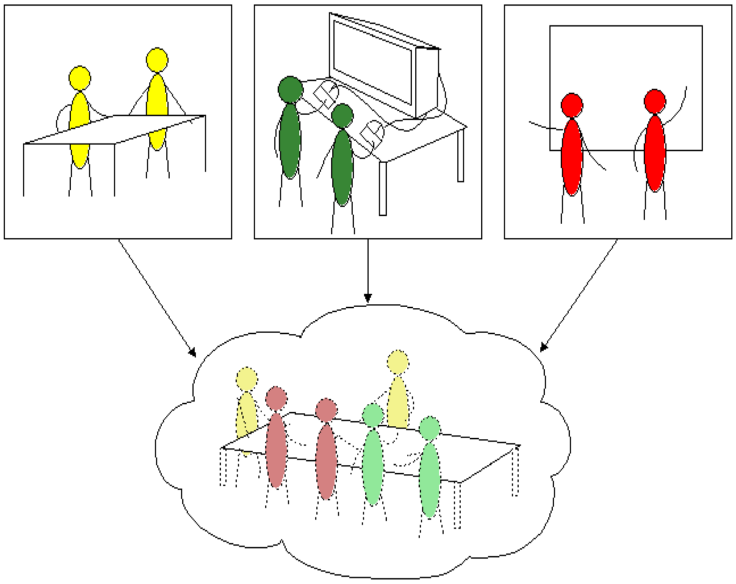


Figure 2.2. Three teams working in a conceptual MPG setting over three connected displays, stylized as a virtual table in the bottom right.

I had two groupware toolkits at my disposal, both developed at the Interactions Lab (my laboratory): the Single Display Groupware Toolkit (SDGToolkit) (Tse & Greenberg, 2002), and the GroupLab Collabrary (Boyle & Greenberg, 2002). Each of these toolkits provides prototyping capabilities for different kinds of groupware. I used these toolkits together to provide the necessary infrastructure for MPGSketch.

The Single Display Groupware Toolkit (SDGToolkit) is a toolkit for rapidly prototyping SDG. It captures and manages multiple mice and keyboards attached to a single computer, and presents them to the programmer as uniquely identified input events relative to the application window. It automatically provides multiple cursors and labels, one for each mouse. To handle orientation issues for tabletop displays, the SDGToolkit automatically rotates the cursor and translates input coordinates so the mouse behaves correctly. The SDGToolkit also provides an SDG-aware widget class layer that significantly eases how programmers create novel graphical components that recognize and respond to multiple inputs.

The SDGToolkit makes it easy to convert conventional single user applications into applications that support multiple users. It is integrated into Microsoft Visual Studio as a graphical component that can be easily dropped into any existing application. Programmers then create an SDG application using familiar programming concepts like object properties and methods, and events and callbacks. Finally, although the SDGToolkit was originally designed for multiple mice and keyboards, it also supports several input devices for large wall or table displays. In sum, the SDGToolkit makes it easy to create single display groupware applications; however, it does not provide support for distributed participants.

The GroupLab Collabrary (Boyle & Greenberg, 2002) is a toolkit that combines easy access to audio and video capture and manipulation with a groupware application data sharing infrastructure. Although it is specifically designed to aid rapid prototyping of media spaces (Bly, Harrison & Irwin, 1993), the Collabrary's simple and generic API makes it easy to consume all or part of its functionality in other kinds of applications. In MPGSketch, for instance, I make use of the groupware data sharing, but not the multimedia facilities. The Collabrary also works as an object-oriented package that can be imported into the Microsoft Visual Studio environment.

The data sharing functionality of the Collabrary is made available through its shared dictionary component. The shared dictionary is a fully-replicated data structure that maps string keys to values of any network-marshallable type (e.g., integers, strings, arrays, bitmaps, and even complex programmer-defined objects). The string keys mimic paths in a typical UNIX-like file system and a simple pattern-matching language, akin to that used with file systems. Together, this permits hierarchical grouping of dictionary entries. As a result, programmers gain the benefits of aggregation and encapsulation of a hierarchical data structure with the ease of access of hash table. In fact, storing or retrieving a value in the shared dictionary is as straightforward for the end-programmer as accessing a value in an array, and is accomplished in a syntactically identical manner.

Updates to the dictionary are serialized through a centralized server architecture and pushed out to clients using a proprietary binary network wire protocol over persistent TCP connections (Boyle, 2003). Thus, the shared dictionary may be considered a notification server (Ramduny, Dix, & Rodden, 1998) with cached keys and values. No separate server software is used: the server and client are implemented in the same object library and servers may be started programmatically just as easily as clients may be connected. End-programmers may subscribe to changes to patterns of keys in the dictionary. With these subscriptions, they receive programmatic notification of additions, modifications, and deletions to entries in the dictionary. The end-programmer can attach a notification event handler to, say, update a GUI with new information stored in the dictionary. With these subscription notifications, the shared dictionary permits distributed groupware development using a distributed model-view-controller paradigm originating from the GroupKit toolkit (Roseman & Greenberg, 1996).

For instance, consider a shared dictionary structure that will maintain the current cursor coordinates of two participants. While there are many ways to do this, one possible structure is shown in Figure 2.3(a). The second field is a unique ID that serves to differentiate people. The local client can set the values of the cursors using the simple calls in Figure 2.3(b). A separate client can *subscribe* to these changes by setting up a subscription (Figure 2.3(c)), and then writing an *event handler* that is automatically called when the values in the keys change (Figure 2.3(d)).

MPGSketch heavily leverages the efforts of both these toolkits (Figure 2.4). The SDGToolkit manages the multiple mice/keyboards attached to each computer, and drawing the local cursors. It assigns each mouse a globally unique identifier and tracks the coordinates of its corresponding cursor. The MPGSketch instance then distributes this data via the Collabrary shared dictionary to other MPGSketch instances running on different computers. It stores mouse identifiers and updates the cursors' on-screen coordinates as they move. Remote MPGSketch instances (using the cursor component of the SDGToolkit) then draw cursors at the correct location for all of the remote input devices listed in the shared dictionary. Finally, as someone draws, the drawing coordinates are also placed in the shared dictionary. Based on this information, the MPGSketch instances update the drawing to give the shared view.

Shared Dictionary Key	Shared Dictionary Value
/participant/1/name	Tony
/participant/1/cursor/X	20
/participant/1/cursor/Y	34
/participant/2/name	Carman
/participant/2/cursor/X	405
/participant/2/cursor/Y	234

Figure 2.3(a). A sample shared dictionary structure to track the cursor locations of two participants.

```
SharedDictionary sd = new SharedDictionary();
sd.Url = "tcp://192.168.1.100:sd";
sd.Open(true);
...
sd["/participant/1/cursor/X"] = 21;
sd["/participant/1/cursor/Y"] = 35;
```

Figure 2.3(b). The first three lines of code connect to a shared dictionary. The latter two lines of code modify the first participant's cursor location.

```
Subscription s = new Subscription();
s.Pattern = "/participant/1/cursor/*";
s.Dictionary = sd;
s.Notified += new SubscriptionEventHandler(cursor1_Changed);
```

Figure 2.3(c). This code sets up a subscription. A subscription is a method of specifying the desire to be notified when changes to certain key/value pairs occur. In this case, the pattern specifies that we are interested in changes to participant 1's cursor position. When changes do occur, we want the method cursor1_Changed to be called.

```
void cursor1_Changed(object sender, SubscriptionEventArgs e) {
      if (e.Path == "/participant/1/cursor/X")
            WriteLine("Participant 1's X cursor position is now "
                  + e.Value);
      else if (e.Path == "/participant/1/cursor/Y")
            WriteLine("Participant 1's Y cursor position is now "
                  + e.Value);
}
```

Figure 2.3(d). This function is called when the values inside the shared dictionary are changed. The function prints out the cursor position when it is called.

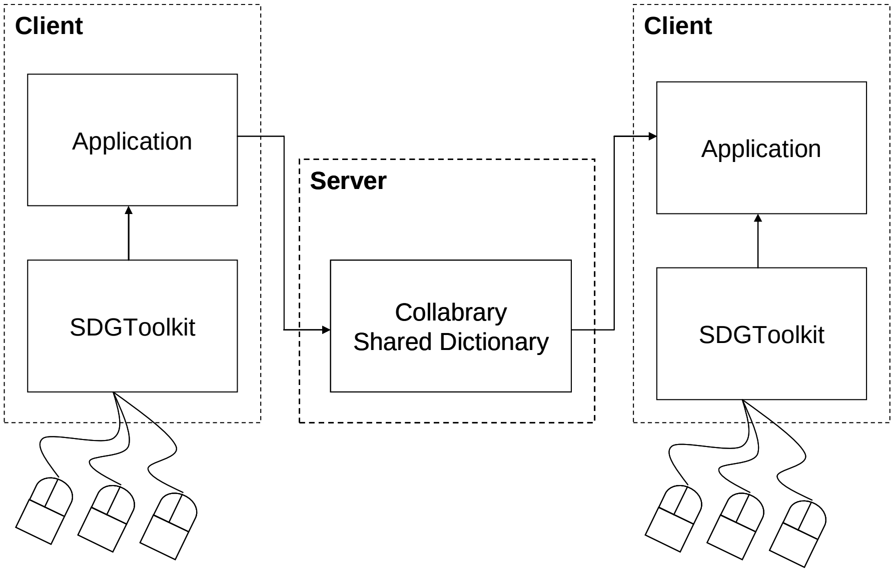


Figure 2.4. Toolkit use in MPGSketch.

2.1.3 An architectural pattern for mixed presence groupware

I have replicated the architectural style of MPGSketch several times in my research, and three of these attempts appear in this thesis. From my experience, it is a suitable architectural pattern for MPG applications since it relies primarily on the underlying toolkits to provide infrastructural functionality. The bulk of my coding efforts have been to build application functionality on top of the toolkits; thus, my prototypes are extremely simple by most programming metrics.

This architectural pattern solves the problem of prototyping simple mixed presence groupware applications. It suggests a method for structuring the application in such a way that allows for the bulk of the programming effort to be placed in the presentation and domain capabilities.

The pattern depends on the use of two programming libraries: one library to provide high-level network communication constructs, and a second library to provide support for multiple input devices. Generally, the more abstract these libraries are, the easier it will be for the end programmer to focus efforts on building the MPG application functionality. Unfortunately, the trade-off is a potentially brittle MPG application whose problems are not diagnosable because of problems in the underlying toolkits. In general though, this is a reasonable trade-off since the pattern is focused on prototyping MPG applications.

For communication constructs, I have generally relied on the GroupLab Collabrary, which, as mentioned earlier, provides a *Distributed Model-View-Controller* (dMVC) architecture (Boyle & Greenberg, 2002). The dMVC architecture is based on the Model-View-Controller pattern, originating from GUI programming, which advocates the separation of the three constructs. A *model* is an object that represents data in the application, and a *view* is a visualization of that data. A *controller* provides a means to change the model, and these changes are implicitly propagated to the view since the view is responsible for having an up-to-date visualization of the model. The dMVC architecture is modeled after this pattern in a network client-server architecture: clients are responsible for controlling and visualizing a model that sits on the server. In my particular MPG prototypes, I implemented the dMVC using the Shared Dictionary of the Collaborary. Other distributed notification servers, such as Elvin (Segall & Arnold, 1997), can be used to implement the dMVC so long as they provide network communication abstractions.

The SDGToolkit provides support for multiple input devices with minimal set up (three lines of code are required to activate the toolkit in a functional manner) (Tse & Greenberg, 2002). I am unaware of any other toolkits that currently provide similar functionality; however, in principle, other toolkits could be used for similar effect.

The solution involves factoring the application into primarily three layers (Figure 2.5): a View layer, a Controller layer, and Network Communication layer. The View layer is responsible for handling and responding to event callbacks from the network communication layer regarding changes to the application model. In the case of the Collabrary, this involves subscribing to a subset of keys which represent the model. For instance, in MPGSketch, these keys would have been those that represented the drawn pixels. The Controller layer is built on top of the toolkit providing the multiple input devices, and is responsible for translating user actions into a set of model changes. In the case of the SDGToolkit, this means writing input device event handlers that understand what changes need to be made to the model. For instance, in MPGSketch, the event

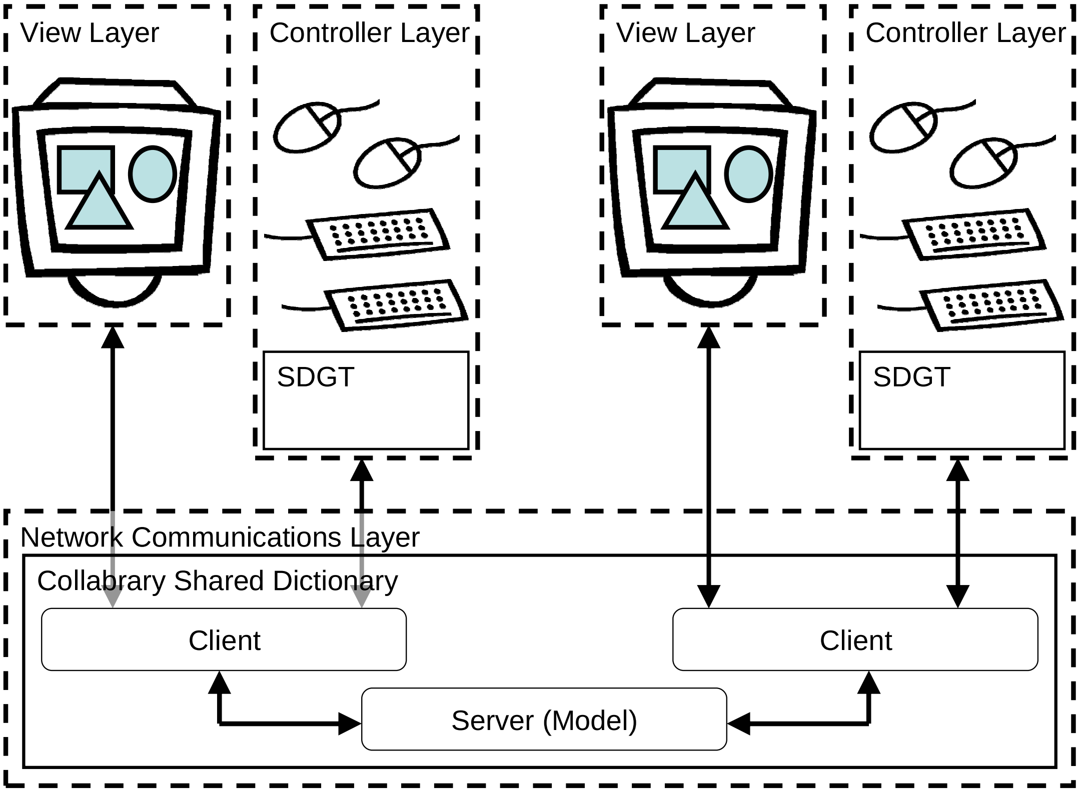


Figure 2.5. Architectural pattern for MPG prototypes.

handlers needed to send off mouse pointer location and drawing changes. These model changes are passed onto the Network Communication layer, which is responsible for actually changing the model and propagating the model changes to listening Viewers. The Collabrary's Shared Dictionary handles this responsibility by effectively sending model changes to the server, which in turn propagates changes to subscribed listeners.

Using this architectural pattern frees the programmer from the details of networking, and input device handling. The key gain of following this pattern is simplicity and time savings. In practice, this architectural pattern worked well for prototyping MPG applications in my laboratory setting, where network latencies and jitter were low and packet loss was non-existent. If the system were to be deployed in the less reliable internet world, the system would have to be refined to address expected latency and jitter problems. However, there is no reason why the underlying pattern would not work in principle.

Finally, while I have depended primarily on the GroupLab Collabrary and the SDGToolkit, in principle, other toolkits providing similar capabilities would suffice.

2.2 Display disparity on heterogeneous displays

Single display groupware tends to focus on large displays (whiteboard or tabletop sized) because of their inherent ability to support co-located groupware. Because I envision MPG to be a vehicle for co-located groups to collaborate across physical distances, I begin by considering a very plausible scenario of MPG use: one site using a large tabletop display while another uses a large whiteboard display. Unfortunately, such a simple scenario raises a number of issues with respect to orientation because of *display disparity*.

This problem arises because vertical displays (such as monitors) have absolute notions of up and down, while tabletops have either an undefined or arbitrary notion of which side is "up." Thus, because this natural affordance of vertical displays is absent in horizontal displays, display disparity raises the following issues:

- How does the system know where users are sitting around the horizontal display?
- How do we mechanically and visually orient pointing devices (e.g. mice) to reflect a participant's seating position? How should this orientation be treated on local displays and remote displays?
- How do we manage "non-upright" orientations on upright displays?
- How do we manage "non-upright" orientations on remote horizontal displays?

In this section, I will describe the *display disparity* problem by first describing how orientation works on digital tabletop displays, then showing how connecting them to traditional upright displays causes problems.

2.2.1 Tabletop orientation

Unlike vertical displays, people can be seated across or at right angles from one another around tabletop displays. Multiple seating arrangements introduce mechanical and visual orientation issues (Kruger, Carpendale, Scott & Greenberg, 2003). Suppose North is the traditional upright location. First, people in a non-North seat will be holding their mice at non-upright angles, which means that coordinates being returned by the devices are incorrect. Second, content (including labeled cursors) oriented correctly for one person will appear sideways or upside down to others. This problem is not particular to MPG—rather, the problem applies more generally to tabletop single display groupware.

Fortunately, the SDGToolkit recognizes tabletop orientation. Each mouse can be associated with a side of the table (and implicitly, an orientation): North, South, East and West. All internal mouse coordinates are transformed relative to that orientation, so that the mouse behaves correctly for the user. Similarly, the labeled cursor is automatically oriented with respect to that orientation. However, the toolkit does not enforce any strategy for content orientation or reorientation—these remain up to the implementer and are open research questions (Kruger, Carpendale, Tang & Scott, 2004).

2.2.2 Heterogeneous orientation

Though much research is focused on strategies for managing content orientation on a single tabletop display, it does not solve the MPG-specific *display disparity* problems of what to do when multiple heterogeneous displays, including horizontal and vertical displays, are connected. For instance, the collaborators in the top-left image in Figure 1.3 view the workspace upside-down or sideways compared to the collaborators working in front of the whiteboard in the bottom-left image in Figure 1.3.

What does it mean to connect vertical monitors with horizontal tabletops? One problem is that we need to establish their relative orientations. As a simplistic solution, we can assume that vertical monitors are always oriented in the North position; we can then arbitrarily assign a table a North position and demand that people work side by side at that position. However, this solution results in “overcrowding” of the North side (see Figure 2.2, bottom right). While it is unclear whether overcrowding is bad, a few reasonable heuristics can help distribute participants around the sides of the virtual table while preserving the physical orientation of co-located users.

1. Users' locations around physical tables are preserved around the virtual table.
2. Users who are seated side by side at an upright display remain seated next to one another at the virtual table.
3. Connected upright displays are automatically placed at different sides of the table.

Regardless of how we "distribute" users around the display, we are left with the problem of how to display other non-upright orientations. For instance, South's cursors and actions will be upside-down to a North individual, while East and West's actions will be sideways (e.g., see Henry and Lothar's cursors in Figure 2.1). While this is expected over tabletop displays, it looks decidedly odd—even unsettling—when this happens on a vertical display. We could translate cursors so they at least appeared right-side up on the vertical display, but this solution would not work for items drawn on the surface that retain their orientation (e.g. text); furthermore, it would be misleading to remote collaborators (cursor orientation implies a collaborator's seating orientation).

If we do not fix orientation, another problem is how people choose "sides" of the virtual work surface. One strategy is to let people do this manually. Another strategy is to have the system assign sides (e.g. to prevent overcrowding of any one side, it may try to balance people around the sides of the virtual work surface). Alternatively, as in the case with all vertical displays, the system may favour a single side to give the majority a common orientation.

In the MPGSketch prototype, mice were positioned around the tabletop with implied orientations. Thus, users could select their personal orientation of the workspace by simply selecting and using a mouse. However, apart from identifying this problem, I chose not to address the problem in my research. My approach for dealing with display disparity in later MPG prototypes was simply to orient the workspace and input devices so that most users would view it with the same orientation.

Solving display disparity remains an open research question. I suspect solutions to this problem to be context dependent, relying on the semantics of orientation in the given context. For instance, orientation of a deck of cards is irrelevant, while the orientation and alignment of elements in a newsletter/publication system are of utmost importance.

2.3 Embodiment and presence disparity in MPG

With the technical infrastructure of MPGSketch in place, I conducted an exploratory study of its use to explore social issues inherent in MPG (Problem 1). To temporarily finesse the orientation issue, I used only upright monitors with common “North” orientation. I placed two pairs of participants (all knew each other well) in front of conventional workstation monitors on either side of a partition. Each workstation ran an instance of MPGSketch and had two attached mice. While people on one side of the partition could not see those on the other side, they could clearly hear them as they spoke. The four people then performed a non-competitive collaborative sketch. While this experimental situation appears suspect—with only one group and an uncontrolled task—it was appropriate for my first foray into MPG use. I was interested in “big effects”—obvious issues, failures and successes—to guide future investigations, and as typical in early testing, these are often seen in even very limited study situations.

I saw no immediately obvious problems associated with group drawing. However, I was surprised to observe that most of the participants’ spoken utterances were directed toward their co-located partners. Rarely, if at all, did participants speak across the partition to the remote group. That is, there was a *conversational disparity* between collocated and remote participants. This disruption to natural conversational dynamics is clearly a major issue, as disruptions to conversational dynamics necessarily disrupt collaborative dynamics. To understand why conversational disparity occurred, I looked into the role of people’s *embodiments* and the differences in *presence* they introduce in collocated-distributed real-time work.

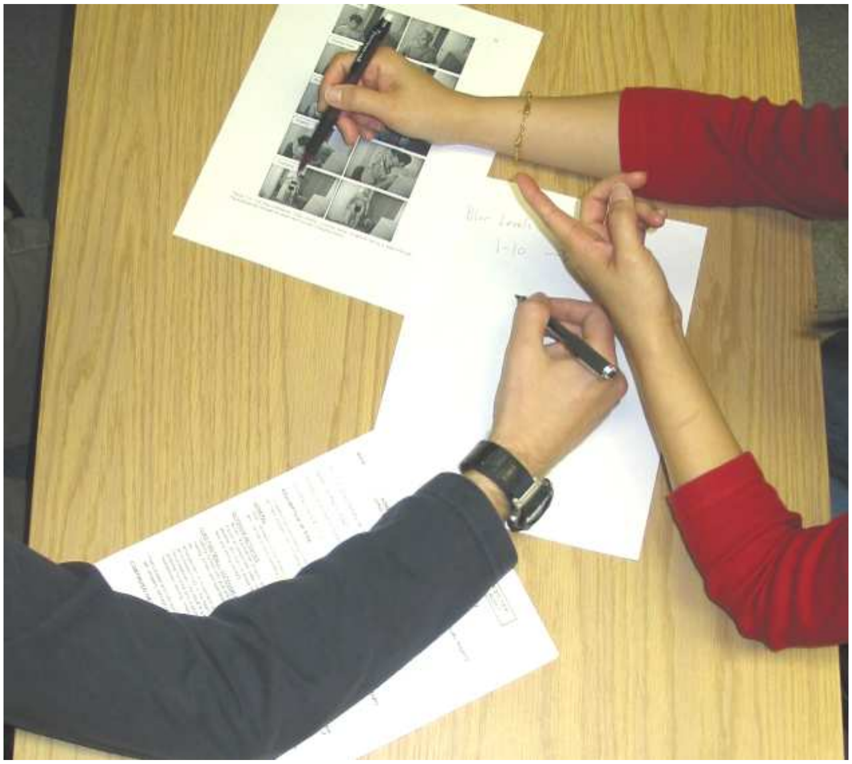


Figure 2.6. Corporeal arms in a common workspace.

As with many real-time groupware systems, MPGSketch represents all remote participants with cursors (or telepointers), as seen in Figure 2.2. In distributed groupware, this small cursor (typically 32x32 pixels) is a remote user's only embodiment in the shared workspace when they are not actively drawing. While cursors are simple, they have proven effective in distributed settings. The presence and movement of the cursor serves as the visual representation of a remote person's presence and activity, and people show remarkable resilience against the missing information, often altering work and conversational strategies in subtle but effective ways.

The problem in mixed presence groupware is that there is a huge disparity between the embodiments of remote people (cursors), and the real-world embodiments of local people (bodies). I call this difference *presence disparity*. For example, contrast people's real world arm embodiments in Figure 2.6 with the cursor embodiments in Figure 2.2. The size disparity alone is a major factor: arms are many orders of magnitude larger than a remote user's cursor, and thus command much more attention. The low information richness and accuracy of the cursor embodiment is another factor.

- Cursors may suggest where its owner is looking but cannot guarantee it.
- An idle cursor (i.e. one that remains stationary for a while) suggests a person's presence, but again cannot guarantee it; in contrast, a person's physical presence is binary: one is either physically present or not.
- The orientation of a cursor suggests where they are seated at a virtual table, but cannot indicate how the person may actually be seated relative to that display in real life. Similarly, the physical orientation of an individual is indisputable.
- Cursor gestures are reduced to referential gestures known as deixis (Clark, 1996); in contrast, bodies can emit emblems and illustrations (Baker et al., 2000; Short, Williams & Christie, 1976), with emblems and illustrations difficult to perform.
- Cursors cannot transmit bodily proximity to others as happens in real life when a person leans in towards another to initiate conversation.
- While people normally initiate computer actions with their mouse, some cursor actions may be too quick or even invisible for others to see. This interferes with others' ability to infer intentions, and to react to them in a timely manner. In contrast, a body is large, and actions take time, allowing others to infer and react to that person's intentions.

Finally, I believe that the presence disparity caused by the embodiment differences lead to the conversational disparity seen in mixed presence groupware. Because co-located embodiments dominate in presence through their size and richness, people direct nearly all of their utterances to co-located collaborators.

2.4 Summary

I show in this chapter that in spite of the inherent complexity of developing MPG software, utilizing toolkits in the architectural pattern I have described for MPG software can considerably ease development efforts (Goal 1). The display and presence disparity problems I have defined in this chapter are particular to MPG systems (Goal 2). The display disparity problem is the idea that when connecting heterogeneous displays (for example, a table display with a whiteboard display), it is unclear how to communicate the fact that collaborators have differential perspectives of the workspace, nor is it clear how to orient objects in the workspace. The presence disparity problem is also unique to MPG systems. It arises because in MPG contexts, some collaborators are collocated while others are remote. Because of this arrangement, collaborative efforts with collocated users are considerably easier than with those who are remote.

The problem of presence disparity now becomes the key focus of this thesis. In the last section of this chapter, I began discussing embodiments as an aspect that differs drastically between collocated and remote participants, and potentially being a source of presence disparity. In the next chapter, I more fully develop the theory of groupware embodiments, drawing from HCI and psychology literature.

Chapter 3. Foundations for mixed presence groupware embodiments

In the previous chapter, I introduced the presence disparity problem that is inherent in mixed presence groupware systems. Presence disparity arises because physically collocated collaborators are seen in full fidelity while remote participants are represented by their embodiments—the virtual presentations of their bodies. Most groupware systems reduce the virtual presentation to little more than a mouse cursor, which clearly cannot compete against the physical body of a collocated collaborator. Thus, presence disparity unbalances the collaborator's subjective experience because even dyadic collaborative dynamics will vary in terms of how one senses presence, engagement and involvement of collocated *vs.* remote partners.

Embodiments are surrogates for remote collaborators. Although we cannot recreate a *physical* body for collaborators in remote locations, what can we do to *virtually* recreate a remote collaborator? More fundamentally, what aspects of a remote collaborator are important to recreate? These questions are summarized by Problem 2: "We do not fully understand the role of embodiments in the specific context of mixed presence groupware."

This chapter addresses Problem 2 by developing a theory of embodiments for mixed presence groupware to answer these questions. I begin developing this theory by examining how physical bodies facilitate collaboration in physical workspaces. I then recast this theory into a set of four design implications for virtual embodiments in MPG. The purpose of these implications is to reduce the disparity between collocated and remote collaborators within MPG. I propose that by building embodiments that address these implications, we can mitigate presence disparity (Goal 2). I demonstrate the utility of these design implications by evaluating the three major classes of embodiments commonly employed today in distributed groupware, and then use the implications as a set of requirements for my embodiment system described in the next chapter.

3.1 Approach

The core problem of presence disparity arises from the physical distribution of participants in the virtual workspace—the physical *presence* of collaborators varies across an MPG workgroup. In my initial informal observations of groups using MPG, I have seen that this presence disparity has negative effects on conversational dynamics (Tang, Boyle & Greenberg, 2004). Because MPG collaborators cannot communicate (verbally and non-verbally) as effectively with remote collaborators as they can with those who are collocated, they will tend to focus their communicative efforts toward their collocated partners (Finn, Sellen & Wilbur, 1997). Remote collaborators are less likely to be invited into informal discussions of the work objects, and are therefore less likely to perform the task as effectively as collocated counterparts.

My approach to mitigating presence disparity in MPG is to understand how the observable aspects of a person's presence play a role in collaboration. For collocated participants, the person's entire body is the observable aspect of a person's presence. However, a body is much richer in communicative value compared to the kinds of groupware embodiments normally seen by remote participants (e.g. telepointers). To appreciate this difference, we need to understand how seeing a collaborator's body influences collaborative work.

Using existing CSCW literature, social psychological theories, and my own experiences with MPG, I present a set of implications for the design of MPG embodiments.

1. To provide feedback of what others can see, one's embodiment should be visible not only to one's distant collaborators, but also to oneself and one's collocated collaborators.
2. To support consequential communication for both collocated and distributed participants, people should interact through direct input mechanisms (like touch

sensitive devices), where the remote embodiment is presented at sufficient fidelity to allow collaborators to easily interpret all current actions as well as the actions leading up to them.

3. To support bodily gestures, remote embodiments should capture and display the fine-grained movements and postures of collaborators. Being able to see these gestures means people can disambiguate and interpret speech and actions.
4. To support bodily actions as they relate to the workspace context, remote embodiments should be positioned within the workspace to minimize information loss that would otherwise occur.

Thus, addressing presence disparity in MPG is reduced to building embodiments that support these properties. I recap literature examining the role of bodies in collaborative work. The context of most of this work is either physical bodies in physical workspaces or embodiments in distributed groupware. As I review the literature, I describe the parallel situation in mixed presence groupware, thereby deriving each implication.

3.2 Bodies in collaborative work

This section reviews three concepts central to bodies in collaborative work: feedback and feedthrough, consequential communication, and gestures. While these concepts are important to distributed groupware systems in general, I recast them as implications for the design of MPG embodiments (Problem 2, Goal 2).

3.2.1 Feedback and Feedthrough

Our ability to perceive our own bodies plays a key role in how we interact with the world (Robertson, 1997). We perceive our own actions and the consequences of our actions on objects as *feedback*, and we constantly readjust and modify our actions as our perceptions inform us of changes to the environment, or changes about our bodily position. Consider the difficulty of threading a needle, and compare that to the difficulty of performing the same task blindfolded. Our ability to perceive our own bodies as physical objects in the world facilitates our smooth interaction with the world.

In distributed groupware, feedback is echoed to other participants as *feedthrough*: the reflection of one person’s actions on other users’ screens (Dix, Finlay, Abowd & Beale, 1998). By observing feedthrough, remote participants can understand a person’s bodily actions and the effect they have on the workspace.

Feedback not only informs the local person of his or her own actions, but gives that person and his or her collocated partners an expectation of what feedthrough is being transmitted (and thus visible) to their remote counterparts. For instance, television news anchors are provided with “monitors.” The monitors display the feedthrough that is being broadcast to television audiences, allowing anchors to reposition themselves, or correct their posture. Similarly, professional singers often have speakers directed toward them during performances, allowing them to hear the collective sound of the group. Singers can therefore readjust their pitch if they are out of tune. Often, when singers are heard singing out of tune, it is because they have not been provided with such monitors.

When feedback and feedthrough are dissimilar, this adds confusion to how local and remote participants experience the interaction. Such mismatches between feedback and feedthrough in groupware typically occur because of network latency and jitter (Gutwin & Penner, 2002). Because the underlying transport layer for the network packets can be jittery (packets arriving in bunches), the remote embodiment may appear to “hiccup” as it traverses the workspace. Other times, when the network layer has completely failed, a remote embodiment may pause indefinitely. Local participants need to be aware if such situations occur, and so, like television news anchors and professional singers, users of mixed presence groupware systems can benefit from “monitors” so they can adjust their actions or the collaborative process accordingly. As well, the groupware may or may not transmit some information. Local feedback *of feedthrough* clearly tells the local person what information is actually being transmitted.

The importance of echoing feedthrough as feedback to local participants gives our first implication for the design of MPG embodiments. *To provide feedback of what others* *can see, one's embodiment should be visible not only to one's distant collaborators, but also to oneself and one's collocated collaborators.*

3.2.2 Consequential Communication

Our bodies are the key source of information comprising *consequential communication*: the information unintentionally generated as a consequence of an individual's activities in the workspace, and how it is perceived and interpreted by an observer (Segal, 1995). A person's activity in the workspace naturally generates rich and timely information that is often relevant to collaboration. For instance, the way a worker is positioned in the workspace and the kinds of tools or artefacts he is holding or using tells others about that individual's current and immediate future work activities.

We see evidence of consequential communication in a wide variety of literature. Segal (1995), in his studies of flight teams, found that pilots spend 60% of their time simply observing the other pilot's console while it is manipulated. Further, he reports that pilots often react smoothly to another's actions without explicit verbal cuing. Similarly, Gutwin & Greenberg (2002) have observed that "participants would regularly turn their heads to watch their partners work" In small group design activities. Tang's (1991) reports of choreographed hand movements can also be understood in terms of consequential communication: by observing others' actions and activities in a shared workspace, one can fairly accurately predict others' future acts or intentions, thereby easily working with or around them. Consequential communication is an important conduit for maintaining awareness of others, allowing us to monitor, understand and predict others' actions in the workspace without explicit action on their part (Pinelle, Gutwin & Greenberg, 2003). The important role of consequential communication in teamwork is illustrated best with an image (Figure 2.6). Even without accompanying text, the position of people's arms, how they relate to each other and the workspace artefacts, and how they are poised to do work tells a rich story of collaborators' presence, engagement and activities.

Consequential communication in MPG fails if people do not have a balanced view of their collocated and remote participants. Physical workspaces allow us to observe individual atomic-level interactions with the workspace, allowing us to predict future activities well. For instance, in a physical setting, such atomic-level interactions might include: a collaborator moving an arm towards a pair of scissors, grappling at the holes of scissors, lifting and grasping the scissors, and finally moving towards an object to be cut. In a virtual setting, the fidelity of the embodiment dictates our ability to observe others. Unfortunately, virtual environments generally tend away from atomic-level interactions, often preferring to represent activities at a coarser level (e.g., the mouse pointer changes into a pair of scissors, or scissors suddenly appear in the empty avatar's hand). This abruptness makes remote participants' actions less predictable. Indirect input devices (e.g. mice, function keys for invoking actions) can even restrict consequential communication in face-to-face computer environments since participants can no longer see how bodies are attached to actions, or how actions are generated (Gutwin & Greenberg, 1998).

One solution to this disparity in MPG is to increase the fidelity of the embodiment representation, which in turn should increase the richness of the consequential communication that is produced. For instance, providing a video of a collaborator grabbing a physical set of scissors would provide remote collaborators with more timely, predictable interactions. Yet this also means that the system must capture appropriate information to generate a rich embodiment, which is directly related to the input mechanism of the system and how this input is connected to bodily actions. For instance, it is far more informative to observe a collaborator physically reaching over to touch and mark up a picture (on a tabletop such as in Figure 2.6 or on a touch sensitive surface) than to watch her cursor embodiment in the virtual workspace move over the picture via mouse input. In the former scenario, because her entire body is involved, it is easier to understand that *she* is the person responsible for the action in the workspace. Furthermore, in the moments prior to her touching the picture, it is easier for us to predict her future actions: she will reach over and grasp a pen, then she will move her body toward the part of the picture she has been talking about, and finally, she will use the pen to make contact with the drawing surface. In contrast, the cursor embodiment loses information: we do not see who it belongs to (although it could be labelled), we do not see her reach for the mouse, nor do we see her raise her finger before a button press, nor do we see where she moves to after she lets go.

Clearly then, the use of direct input mechanisms (e.g. touch sensitive surfaces) provide collaborators with richer consequential communication than mechanisms providing indirect input (e.g. mice).

If collaborators are to successfully maintain an awareness of distributed participants in MPG workspaces, then their embodiments need to be capable of providing a comparable fidelity and range of expressiveness as physical bodies. Similarly, if we are to mitigate presence disparity, we need to recognize that collocated collaborators will unconsciously use all available consequential acts to communicate ideas with one another; when distributed collaborators cannot see these consequential acts, the entire group's effectiveness suffers.

This brings us to our second implication for the design of MPG embodiments. *To support consequential communication for both collocated and distributed participants, people should interact through direct input mechanisms, where the remote embodiment is presented at sufficient fidelity to allow collaborators to easily interpret all current actions as well as the actions leading up to them.*

3.2.3 Gestures

While consequential communications are unintentional body acts, *gestures* are intentional bodily movements and postures used for communicative purpose (Gutwin, 1997). Gestures play an important role in facilitating collaboration by providing participants with a means to express their thoughts and ideas both spatially and kinetically, reinforcing what is being done in the workspace and what is being said. Gestures are a frequent consequence of how bodies are used in collaborative activity: Tang (1991) observed that 35% of hand activities in a physical workspace were gestures intended to engage attention and express ideas. Because intentional gesturing is so frequent, hindering that communicative process—by not giving participants the ability to view or to produce gestures effectively—may negatively impact collaborative activities in MPG.

Two classes of gestures facilitate the communication of ideas and therefore group work in the shared workspaces: those that are purely communicative acts, and those that relate to the workspace and its artefacts. *Pure communicative gestures* arise from a person's natural communicative effort, and can occur independently from the workspace. People use gestures to facilitate speech production (Krauss, Dushay, Chen & Rauscher, 1995), to emphasize parts of speech and to attract attention (Bekker, Olson & Olson, 1995). Psychological theory suggests that spatial and kinetic gestures are part of people's semantic encoding of ideas, and therefore that the retrieval of words depend on gestures (Krauss et al., 1995). Two key pieces of evidence support this position: first, most gestures appear prior to the accompanied speech, and second, preventing speakers from using gestures tends to impede smooth speech production. For instance, Morrel-Sammuels & Krauss (1992) found that gestures usually precede speech by a 0.75s interval. More telling is that speakers' fluency has been found to be markedly hampered when they are prevented from gesturing (Rauscher, Krauss & Chen, 1996). Listeners also use accompanying gestures to interpret and disambiguate speech. Riseborough (1981) found in two separate experiments that participants benefited drastically when able to view accompanying gestures compared to speech alone, both in terms of word recall and recognition. Some gestures, called emblems (Short, Williams & Christie, 1976), also convey semantic information above and beyond speech alone, and may replace speech entirely (e.g. yes or no via thumbs-up or thumbs-down, insults via the middle finger). At a higher level, gestures are also used to help regulate conversation (Bekker et al., 1995). For instance, people use gestures to negotiate turn-taking. such as putting up your hand to express a desire to speak, or gesturing at the person who can speak next (Duncan, 1972).

Workspace-oriented gestures are the class of gestures that directly relate to the collaborative workspace and the artifacts within them. Deixis refers to gestures that refer to objects or locations in the workspace (Clark, 1996), while *illustration* clarifies verbal communication over the workspace (Harrison & Minneman, 1994). Of course, there are many types of workspace-oriented gestures and they can be used for many different things. Bekker et al. (1995) developed a taxonomy of gestures from observations of ten different teams performing collaborative design over a workspace. First, they identified four different *types* of gestures, of which three are workspace-oriented:

- *Kinetic*: movement that illustrates an action sequence.
- *Spatial*: movement that indicates distance, location, or size.
- *Point*: fingers point at a person, object, or place. The target may be concrete, abstract, denoting an attitude, attribute, affect, direction, or location. This type of gesture is often referred to as a deictic reference.

Next, they observed that gesture types are often combined into *sequences* (Bekker et al., 1995). For example, one common sequence comprising a workspace-oriented gesture is the *walkthrough:* a succession of kinetic gestures illustrating how something might be used. Another sequence is the *list*, a string of pointing gestures in concert with speech referring to a numerical or bulleted list. Collaborators often combine atomic-level gestures in novel combinations to express ideas; thus, even if an exhaustive taxonomy of atomic gestures was developed, attempting to support remote interaction by providing "canned" gestures would be an insufficient approach.

Finally, Bekker et. al. (1995) determined that gestures have several primary *roles* within a design setting. The one most relevant to workspace-oriented gestures is the *design role*, where gestures relate to the current design activity and refer to things like showing distances, enacting the interaction between user and product, referring to objects, persons or places, etc. The design role is of particular interest to MPG embodiments because it emphasizes that a gesture's semantic information is often tied to the context in which it is produced. For instance, gestures in the workspace often refer to objects or locations on the workspace (e.g. "We should move that one way over here.").

Clearly, people regularly use many different kinds of detailed communicative and workspace-oriented gestures. This leads to our third implication for the design of MPG embodiments: *To support bodily gestures, remote embodiments should capture and display the fine-grained movement and postures of collaborators. Being able to see these gestures means people can disambiguate and interpret speech and actions*.

The above theories also confirm the importance of the relation between gestures and workspace artefacts. Yet the vast majority of distributed groupware separates the visuals of the person from the workspace. As in Figure 3.4 (page 42), usually the person is captured as a video stream and displayed in one window, while the workspace is shown in a different window (e.g. Bly & Minneman, 1990). Even though hand gestures may be visible on the video, they are completely decoupled from the workspace. By virtue of being *about* the workspace or objects on the workspace, removing these gestures from the context of the workspace removes much of the meaning conveyed by them. Indeed, many gestures and accompanying speech may make no sense whatsoever if removed from the context of the workspace (e.g. "This piece should be this big," necessarily raise the questions of, "Which piece?" and, "How big?"). Thus, the fourth implication for the design of MPG embodiments is that: *To support bodily actions as they relate to the workspace context, remote embodiments should be positioned within the workspace to minimize information loss that would otherwise occur.*

This discussion of gestures also reinforces the second implication. Since the ability to freely use gestures is important for fluent speech production, smooth interaction in MPG is necessarily best facilitated by un-tethered input devices, allowing people to work freely and directly over the work surface (e.g., as on a touch sensitive display). Tethering users to input devices (such as a keyboard and mouse) inhibits users from gesturing as part of their communicative effort, hindering vocabulary use and the articulation of ideas (Rauscher et al., 1996).

In summary, I have described three concepts central to how observable aspects of bodies contribute to collaborative work: feedback and feedthrough, consequential communication, and gesturing. I caution that this list is not complete. I have deliberately confined my examination of bodies to the observable aspects of the body from a "bird's eye-view" of the workspace to keep the problem space of Problem 2 tractable. A body has many other features that contribute to collaboration: for instance, eye contact plays a role in inter-personal communication, and gaze awareness is important for knowing where others are focusing their attention (Ishii & Kobyashi, 1993; Vertegaal, 1999). However, the three concepts I have investigated present several key design implications for embodiments in MPG (Goal 2). By approaching the design of MPG embodiments with these properties in mind, we can build embodiments that mitigate presence disparity.

3.3 Evaluating groupware embodiments for MPG

I now use these four design implications as a basis for understanding the most popular embodiments found in existing distributed groupware and to re-examine these approaches for their suitability within MPG.

In face-to-face situations, we watch others' bodies, their facial expressions, and the workspace to maintain workspace awareness (Gutwin & Greenberg, 2002). In typical distributed groupware systems, we rely on embodiments to represent others so that workspace awareness information can be acquired and maintained (Benford, Greenhalgh, Bowers, Snowdon & Fahlén, 1995). Three approaches have dominated embodiment design in groupware: telepointers, avatars, and video embodiments. All have achieved reasonable success in distributed groupware, for they add information of varying richness where none existed before. Unlike MPG, embodiments in distributed groupware do not introduce imbalance because all collaborators see each other only through the embodiment.

3.3.1 Telepointers

Telepointers (Figure 3.1) are the simplest approach for supporting embodiment, and were implemented as early as 1968 (Engelbart & English, 1968). Remote participants are represented in the workspace as mouse cursors, one for each person. As with the local cursor, mouse movements by participants are shown in real time as movements of corresponding pointers. This visual cue is surprisingly effective in conveying a wealth of information, such as presence, location, movement, selective gestures, and activity. Telepointers can provide implicit identity during speech, since observers are good at associating a telepointer's actions with a speaker's speech.

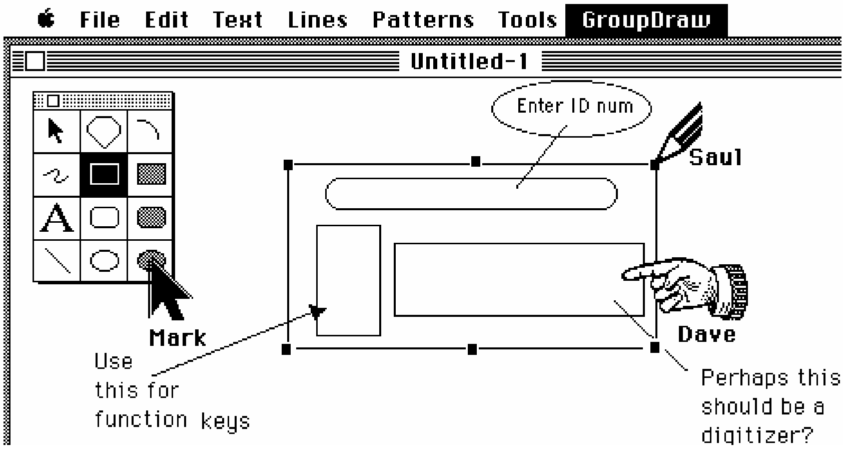


Figure 3.1. Three different telepointers in a groupware application. In this case, each telepointer is labelled and drawn differently, either to denote identity or convey mode information (Greenberg, Gutwin & Roseman, 1996).

Telepointers can provide more information through judicious use of labeling, color, iconography and visual overloading (Greenberg, Gutwin & Roseman, 1996):

- *identity information*, where the owner's identity is made explicit, e.g., by attaching a textual name, a photo or even an abstract symbol to the cursor (Figure 3.1);
- *action information* that reflects and emphasizes its owner's actions, e.g., the rapid selection of an item is emphasized by presenting a miniature mouse with its button shown pressed as part of the cursor visuals (Figure 3.2);
- *mode information* that reflects the owner's interaction state for moded interfaces, again by changing the cursor visuals (Figure 3.1);
- *trace information*, where a visual trace of the telepointer movement over time informs where the cursor has been in the recent past (Gutwin & Penner, 2002).

In spite of their success, telepointers are limited embodiments. A telepointer provides only limited space to incorporate extra information—overloading or rapidly changing its visuals to show pointing, identity, activity information and mode can quickly make it over-cluttered and difficult to interpret. Remote people cannot reliably interpret an idle telepointer (has its owner stepped out for the moment, or is the owner there but not active).

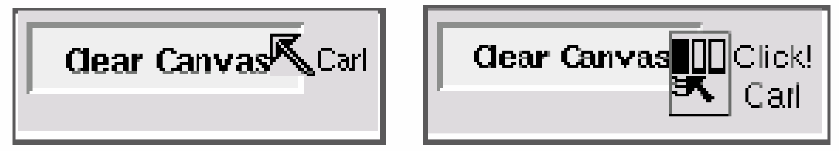


Figure 3.2. Action information can be conveyed by changing the representative icon. In this case, Carl clicks the button on the left, and that information is conveyed to remote participants on the right (Greenberg, Gutwin & Roseman, 1996).

Finally, telepointer gestures are generally restricted to pointing; kinetic and spatial gestures are hard to perform (Gutwin & Penner, 2002).

Within MPG, the added problem is the huge discrepancy between how a person sees the telepointer of their remote collaborator *vs.* the full body actions of her collocated collaborator. The telepointer captures and presents only a fraction of body actions. It is also very small, and thus cannot create the same degree of presence when compared to a body's visual salience. This discrepancy negatively impacts the effectiveness of consequential communication in telepointer-only situations.

3.3.2 Avatars

Avatars (Figure 3.3) originated in collaborative virtual environments (Benford et al., 1995), three-dimensional worlds with immersive input/output systems such as CAVE's, but are now mostly seen in collaborative games. Avatars often appear as humanoid, three-dimensional beings, typically with a distinctive head, body and arms. The idea is to have fairly distinctive human-like representations in what might be considered a three-dimensional simulation of the real world.

The typical avatar portrays only limited information: the location of a person in a space, and roughly where they are looking. While motion through the space is transmitted, most avatars are rendered with poor fidelity and infrequently updated, so seeing or interpreting fine-grained motion is impossible. In addition, avatars are typically abstract or pseudonymous caricatures, making identity difficult to determine. They may not even have arms, which means gesturing is effectively impossible.

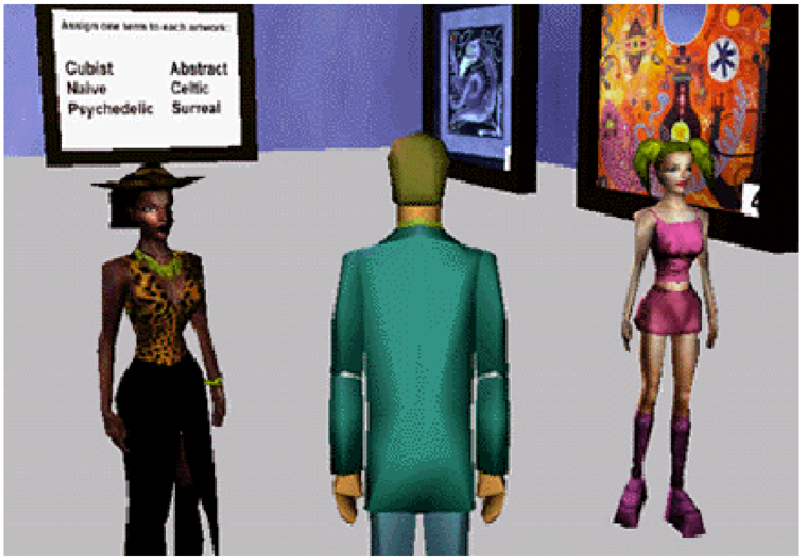
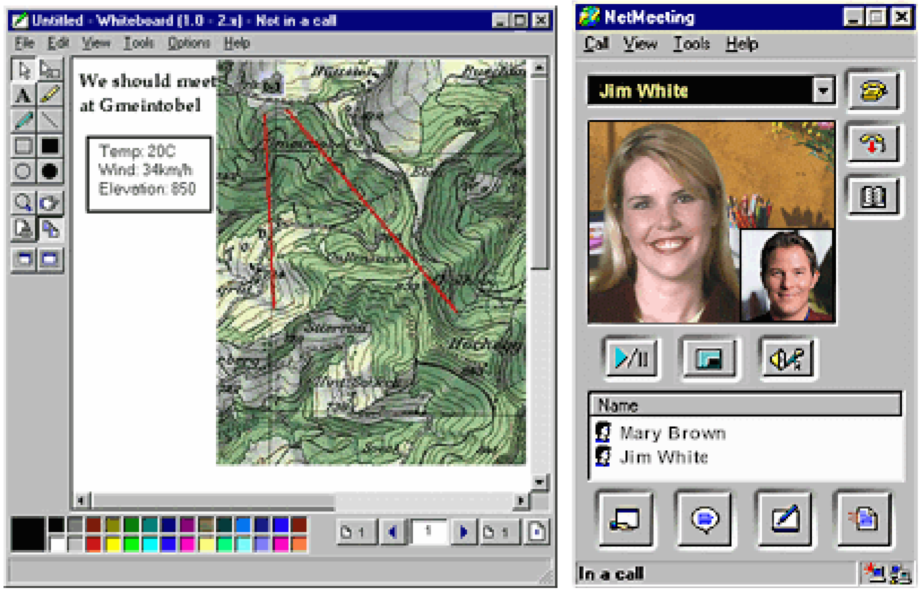


Figure 3.3. Three different avatars in a virtual world (Gerhard, Moore & Hobbs, 2001).

Some avatars portray rich information. For instance, some allow people to animate their avatar’s hand and body positions through canned gestures. Tracking where the person is looking into the virtual space and adjusting the gaze of the avatar to point to the same direction provides others with a sense of where the user is looking (Vertegaal, 1999). To show identity, some systems replace the avatar head with a live video feed of its owner, thus revealing the avatar owner and facial expression (albeit at low fidelity and frame rate) (Vertegaal, 1999). People can also customize their avatars to have a more recognizable face, or to dress them with identifying clothing. Games have made significant headway in this area, providing a vast array of clothing and other bodily enhancements so that many different characters can be distinguished. Interestingly, however, even these avatars are typically displayed with a nametag.

Avatars are a limited means to portray bodies in collaborative settings. While activity is carried out with the “hands” or “arms” of the avatar, only larger actions can be interpreted—the low fidelity of activity representation puts to question the utility of avatars to convey rich consequential communication. Natural gestures are weakly supported by avatars, and are hampered by the poor expressiveness of their controlling devices (mice or joysticks)—typically only canned gestures are supported by keyboard-invoked waves or



We should meet at Gmeintobel

Temp: 20C
Wind: 34km/h
Elevation: 850

Jim White

Name
Mary Brown
Jim White

Untitled - Whiteboard (1.0 - 2.x) - Not in a call
File Edit View Tools Options Help
NetMeeting
Call View Tools Help
In a call

Figure 3.4. A screenshot of Microsoft NetMeeting, showing the worksurface completely separated from images of the collaborators. (Adapted from www.microsoft.com/windows/netmeeting.)

smiles. While data gloves and suits can fix this, they tend to be the exception rather than the rule. Another problem is that of scale: visually, avatars are used in landscape or birds eye-view virtual “worlds” in contrast to the size of the work surface of interest, which is typically closer to a tabletop in size. Thus, they display only gross actions and movements are supported as opposed to fine, granular movements. Finally, and as with telepointers in MPG, the fairly low fidelity avatar representation must compete with how one sees the full body of the local person—a daunting task.

3.3.3 Video overlays

Finally, the use of video as an embodiment technique has been gaining popularity in both consumer-based applications (e.g. MSN Messenger), as well as in the research community (e.g. Apperley, McLeod, Masoodian, Paine, Philips, Rogers, & Thomas, 2003; Tang & Minneman, 1991a; Tang & Minneman, 1991b). The typical teleconferencing system completely separates the embodiment from the work surface (Figure 3.4); however, many research-based video-based teleconferencing systems use two cameras per site, one to capture a person’s face, and the other to capture the workspace (Figure 3.5). People can

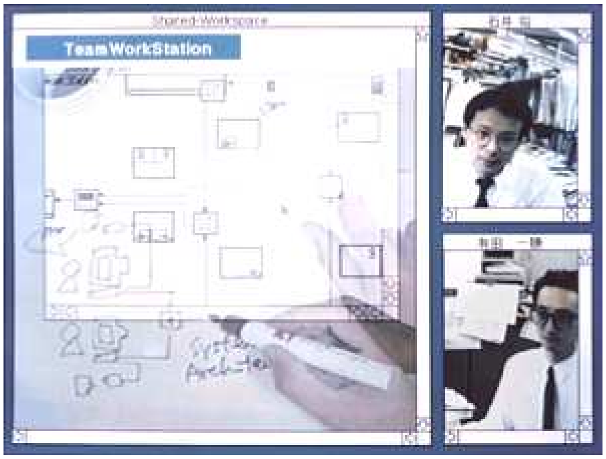


Figure 3.5. In TeamWorkstation, separate cameras capture the workspace and the collaborators (Ishii, 1990).

view these two video streams usually by switching between them, or using picture-in-picture. The workspace stream lends itself to a restricted form of embodiment, since the camera captures and transmits the local person's arms as she works atop it (e.g., when the camera points down to a tabletop). The catch is that video systems like these are one-way. They do not present a shared workspace as the distant person can only see the other's workspace and interactions, but cannot work within it.

This inability to see but not interact with the distant workspace proved frustrating to several architects working within a media space system developed by Xerox PARC (1987). Their solution was to tape tracing paper atop the display of the remote workspace, and to point the local camera to this mixed paper/monitor setting. This "fused" the local and remote workspace into a single view: the camera captured the local person's arms and the marks they made on the tracing paper, as well as the remote person's arms and activities just visible through the translucent paper. Perhaps most importantly, it allowed participants to see the bodies and faces of local and remote participants *within the context* of the shared workspace.

This innovation led to several research efforts on fusing video-based workspaces. First was VideoDraw (Tang & Minneman, 1991a), a video-based solution that used multiple cameras to capture the desktop, and polarizing filters to manage video feedback

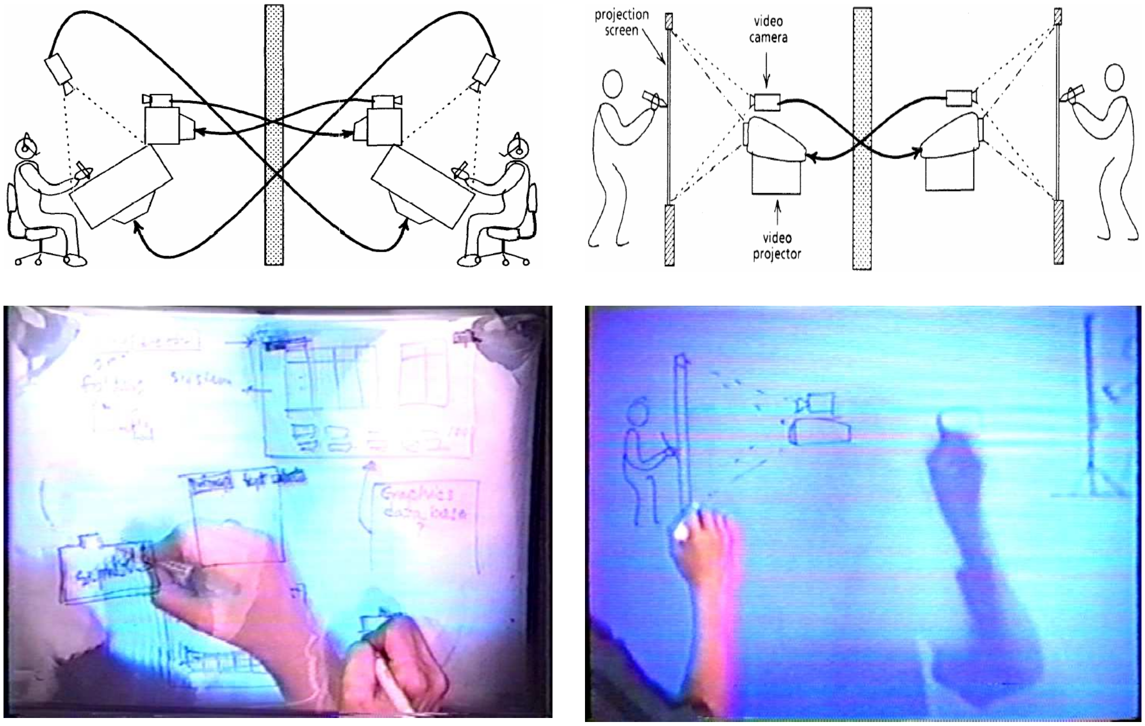


Figure 3.6. The two images on the left show how VideoDraw works (Tang & Minneman, 1991a). On the right, the two images show how VideoWhiteboard works (Tang & Minneman, 1991b).

(Figure 3.6, left). The resulting fused image, showing both the local and remote participants' arms in the space, is exemplified in Figure 3.6, bottom-left. We see that as a video embodiment, VideoDraw allows a full range of fairly sophisticated gestures by giving participants a 2 ½ dimensional gesturing space (three dimensions are flattened into one, but depth cues are preserved) of each other's actions.

Using a similar technique, VideoWhiteboard (Tang & Minneman, 1991b) allowed people to draw on translucent large screens with markers, while a camera mounted behind the screen captured both the drawings and the shadows of people near the screen (Figure 3.6, right). These shadows were seen as silhouettes in the fused video display (Figure 3.6, bottom right), giving the illusion that remote collaborators were on the other side of the screen. The downside is that shadows flatten the gesturing body parts to two dimensions, reducing the range of possible gestures and compromising the interpretation of detailed actions. For example, an "A-OK" sign (thumb and forefinger in a closed circle) may be

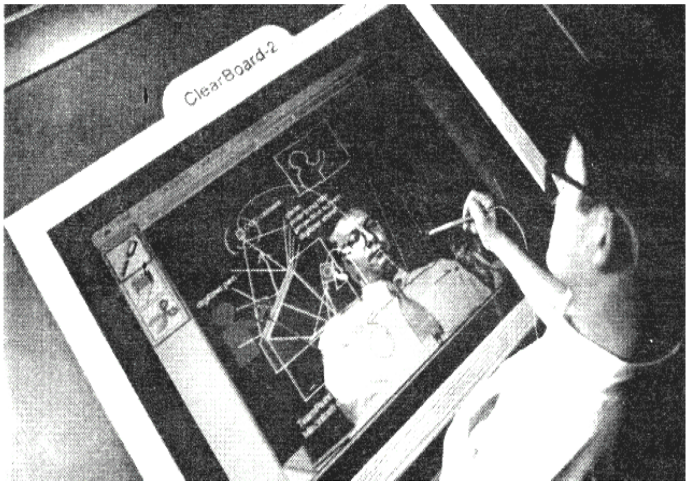


Figure 3.7. ClearBoard seamlessly integrates video and the workspace (Ishii & Kobyashi, 1993).

seen only as a black blob because the shadow will also include the fingers behind the two front ones unless the camera angle is just right. Similarly, arm actions in front of the body may be masked by the shadow of the body itself. Shadows also hide identity, which is problematic in MPG since there is more than one person per site.

Ishii's TeamWorkstation (1990) used video-mixing technology to fuse the different video layers as overlays (see Figure 3.5, where hands appear on the workspace). Unlike VideoDraw and VideoWhiteboard, cameras could point to and fuse otherwise unrelated surfaces, such as a physical desk or a control panel. Ishii then developed ClearBoard, which used half-silvered mirrors and multiple cameras to mix a remote participant's face into the shared video workspace in a way that maintained gaze awareness (Ishii & Kobyashi, 1993). As seen in Figure 3.7, the metaphor is of two people working on different sides of a pane of glass, where each can mark atop their side of the glass with marking pens.

While all the above systems are extremely good at capturing and transmitting live embodiments, they are limited because they are based solely on analog video technology. First, because these systems combine all video frames into one, they degrade substantially if more participants (and video feeds) are added. Secondly, while people can see each other, they cannot manipulate the marks and artefacts created or held by others. A later

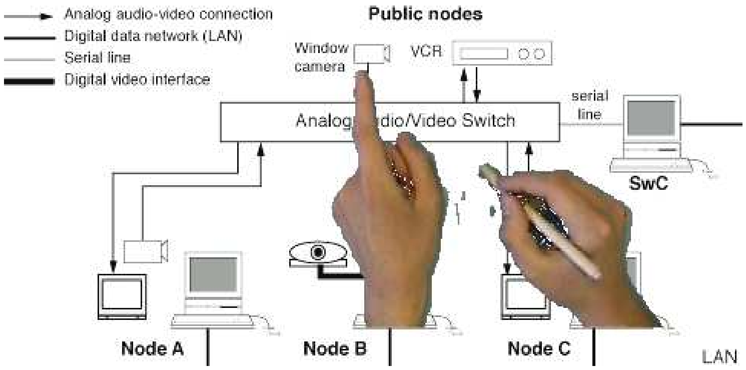


Figure 3.8. Using the hand as a telepointer, from Roussel (1991).

version of ClearBoard finesses this problem by incorporating a see-through digital display showing a groupware system and a digitizing pen for input (shown in Figure 3.7) (Ishii & Kobyashi, 1993), allowing people to share their electronic interactions and artefacts.

To solve this analog/digital fusion problem, Roussel (2001) has people use their arm over a solid blue surface. He extracts these arms by chroma-keying, and then super-imposes them over a digital workspace as a semi-transparent image. This gives a very crisp effect, as seen in Figure 3.8 (Roussel only simulated the groupware capabilities, but there is no technical reason why it could not be implemented). People could also control the properties of these hands as they appeared in the workspace: their size, their relative position, and their transparency. The downside is that people must use their arms outside the workspace, although they can control what they do through the feedback that appears on the display. While reasonable for distance collaboration, such a scheme would likely be confusing to collocated collaborators in MPG.

Finally, LIDS recreates VideoWhiteboard as a digital system that enhances distributed Powerpoint presentations (Apperley et al., 2003). They capture the image of a person working in front of the shared display via consumer-grade cameras, and transform it via background subtraction and posturizing techniques into a frame containing the digital shadow (Figure 3.9). They then overlay three transparent windows to create the scene: the digital shadow, the Powerpoint frame, and a frame that captures sketching overlays. The Distributed Designer's Outpost also includes a shadowing capability captured by rear-

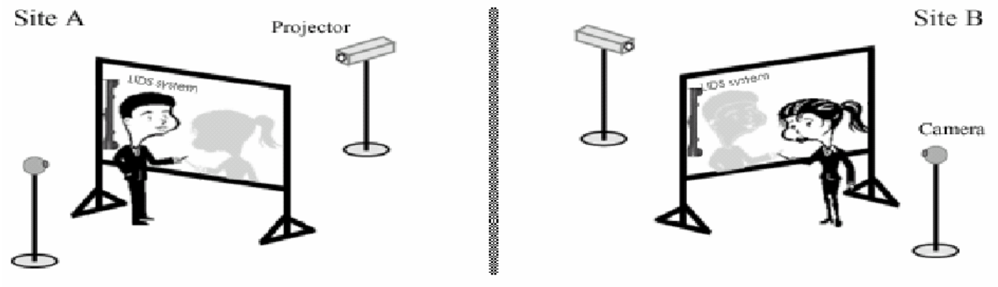


Figure 3.9. LIDS uses a similar approach to VideoArms, but instead of colour segmentation, uses a posturization filter to segment out the shadow of the collaborator.

projection (Everitt, Klemmer, Lee & Landay, 2003). The drawback of both of these approaches is that the resulting shadow cannot convey depth information, limiting the richness of the embodiment. Furthermore, the fidelity of the embodiments is so low that they are only useful for coarse gestures (in the case of LIDS), and for indicating presence (in DDO).

3.4 Another look at video-based approaches

Compared to the physical body, particular embodiment approaches in distributed groupware are clearly lacking in several areas, especially when applied to an MPG setting. I now discuss each of the embodiment techniques in terms of our design implications for MPG embodiments.

The first design implication suggests that a person's embodiment should be visible not only to his distant collaborators, but also to himself and his collocated collaborators. Telepointers and avatars are typically visible by all collaborators; thus local collaborators can see how and what actions are presented to remote collaborators. Video-based embodiments are sometimes but not always visible by all collaborators (e.g., VideoWhiteboard and LIDS do not provide local feedback); the negative consequence of analog video feedback loops can make this hard to do in particular configurations. Still, all three approaches are potentially amenable to present MPG embodiments to both local and remote people.

The second design implication addresses the need to support consequential communication by using direct input mechanisms and through high fidelity MPG embodiments. Telepointers perform poorly because they are typically controlled by indirect input devices, and because they presuppose a limited way for users to interact with the system (pointing and clicking with a mouse). Thus, they present only a fraction of body actions to both local and remote participants. Avatars also fall short: most only represent activity at a coarse, high level and, excepting those controlled by data gloves or suits, also suffer from being controlled by indirect input devices. Video-based embodiments are the most promising. People use their hands and bodies to directly work within the workspace. They are able to provide rich details about the collaborators, especially when full fidelity views (vs. shadows or silhouettes) are used.

The third design implication speaks about the necessity for embodiments to capture and display the body gestures of collaborators. The telepointer limits us too severely to adequately support all gestures, as they are restricted to motion and pointing primitives. Avatars as traditionally implemented are too coarse-grained, leaving them less than ideal. Again, full-fidelity video-based embodiment approaches are the most promising, although we have to be wary of shadow-based approaches that can mask certain gestures.

The fourth design implication stresses that embodiments should be placed within the context of the workspace. Telepointers and avatars only do this partially: while they show some actions in context, these are not connected to the owner's body that might appear in a separate video window and out of context. Recall that being able to see others' bodies as they act facilitates collaboration; if the embodiment, or virtual body, has only a weak link to the physical body, then the utility of the embodiment for collaborative work is compromised. Video-based embodiments, if properly calibrated to the work surface, tightly couple the embodiment within the workspace.

In summary, I believe that video-based embodiments are the most promising approach for MPG because of their ability to capture and convey the rich gestural and consequential communication that is important in collaborative work. Yet video-based embodiments for MPG are currently problematic: analog approaches are costly (cameras, projectors and transmission bandwidth requirements), and overlaying analog video compromises scalability and image clarity. For the non-computer vision specialist, digital image processing has algorithm complexities and performance issues that arise during attempts to extract, manipulate and overlay high-quality images from a noisy scene. For both, the setup, registration and calibration of equipment so that images appear in the correct place are a problem. Another problem is that the promise of video embodiments in MPG is shown by the collective properties of the various systems discussed previously, but none are designed for MPG or satisfy these implications.

3.5 Summary

Using social psychological and CSCW literature, I used this chapter to develop the theory behind mixed presence groupware embodiments (Goal 2). My focus was on three concepts that describe the role bodies play in collaboration: feedback and feedthrough, consequential communication, and gestures. I considered each of these three concepts in the context of mixed presence groupware, and derived four implications for the design of MPG embodiments. I used these implications to evaluate the three primary classes of distributed groupware embodiments today, and found that a video-based approach holds the most promise for mixed presence groupware. With the four implications in hand, I now attempt to address presence disparity by building an new embodiment technique called VideoArms.

Chapter 4. VideoArms: An MPG Embodiment System

Chapter 3 laid the groundwork for mixed presence groupware embodiments by articulating four principles that MPG embodiments should fulfill to mitigate presence disparity. In this chapter, I put these principles to practice by building a prototype embodiment system called VideoArms (Goal 3). I motivate VideoArms by describing an early mockup of the system that did not use video. I use that experience to describe the concept behind the VideoArms embodiment system, and then describe its functionality. Finally, I dive into the implementation details of VideoArms, weighing the ramifications of different technical approaches.

4.1 Why are arms good embodiments?

When I began looking at embodiments, I started by looking at videos and images of people working over physical work surfaces (e.g. Scott, Carpendale, & Inkpen, 2004), comparing them to my experiences with standard groupware whiteboard applications. One of the things I noticed immediately was the prominence and sheer size of the body in physical workspaces, especially when contrast against the typical telepointer embodiment found in most groupware drawing systems.

Even at a very high level, the arm (the visible aspect of a body when a workspace is viewed from above) exhibits a number of characteristics that are typically not conveyed properly by a telepointer. For a more detailed analysis, I took key components of a framework for comparing embodiments in distributed groupware (Benford et al., 1995), where I used it instead to compare corporeal arms (physical arms) with telepointers.

1. ***Presence***. The presence of an individual's body (in particular, his or her arm) in the workspace, indicates the person's presence. When the individual leaves, the body goes with the person. There is no ambiguity about a person's physical presence. In contrast, typical groupware systems equate a network connection with "presence": the telepointer is displayed so long as the connection is maintained. As mentioned earlier, the visibility of a telepointer is decoupled from the physical presence of a collaborator: telepointers remain visible even when a collaborator leaves the room.

2. ***Grounding of gestures and person-specific orientation***. A corporeal arm is necessarily attached to a body, and so even from an overhead view, the visible aspects of the arm ground gestures with respect to a specific individual. In addition, the position of the arm over the workspace conveys a person-specific orientation. By conveying distinct, person-specific orientations, drawings by each participant can be interpreted correctly (Kruger, Carpendale, Scott & Greenberg, 2003). For instance, Tang (1991) noticed that drawings oriented toward its creator tend to be personal, while those oriented towards others tend to be public. In contrast, a telepointer is small, meaning telepointer actions are less obtrusive, and is less attention grabbing—especially on a large work surface.

3. ***Awareness***. An arm's actions, by virtue of its sheer size, command far more attention compared to actions by a small telepointer. Furthermore, arms obscure the workspace underneath, thereby attracting attention. In contrast, a telepointer is extremely small, meaning telepointer actions are inherently harder to detect—especially on a large worksurface.

4. ***Identity***. Finally, people have extremely varied physical appearances—body/face size, shape and proportion, skin colour, hair, clothes, etc.—that are the essential cues for identity. Because the arms are also grounded to a particular person, the identity of the individual is clear. Although telepointers can be augmented to provide more identity information, deciphering a telepointer's owner is clearly more difficult than finding the owner of an arm.

Even without considering other embodiment aspects crucial to communication (gestures and consequential communication as presented in Chapter 2), corporeal arms clearly outpace telepointers in many aspects that are considered important for embodiments in distributed groupware. I consequently considered what it would be like to have an embodiment that exhibited the simple characteristics of corporeal arms listed above. I was interested, in particular, about changing the presentation of telepointer information to a "telepointer arm" that emulated corporeal arms.

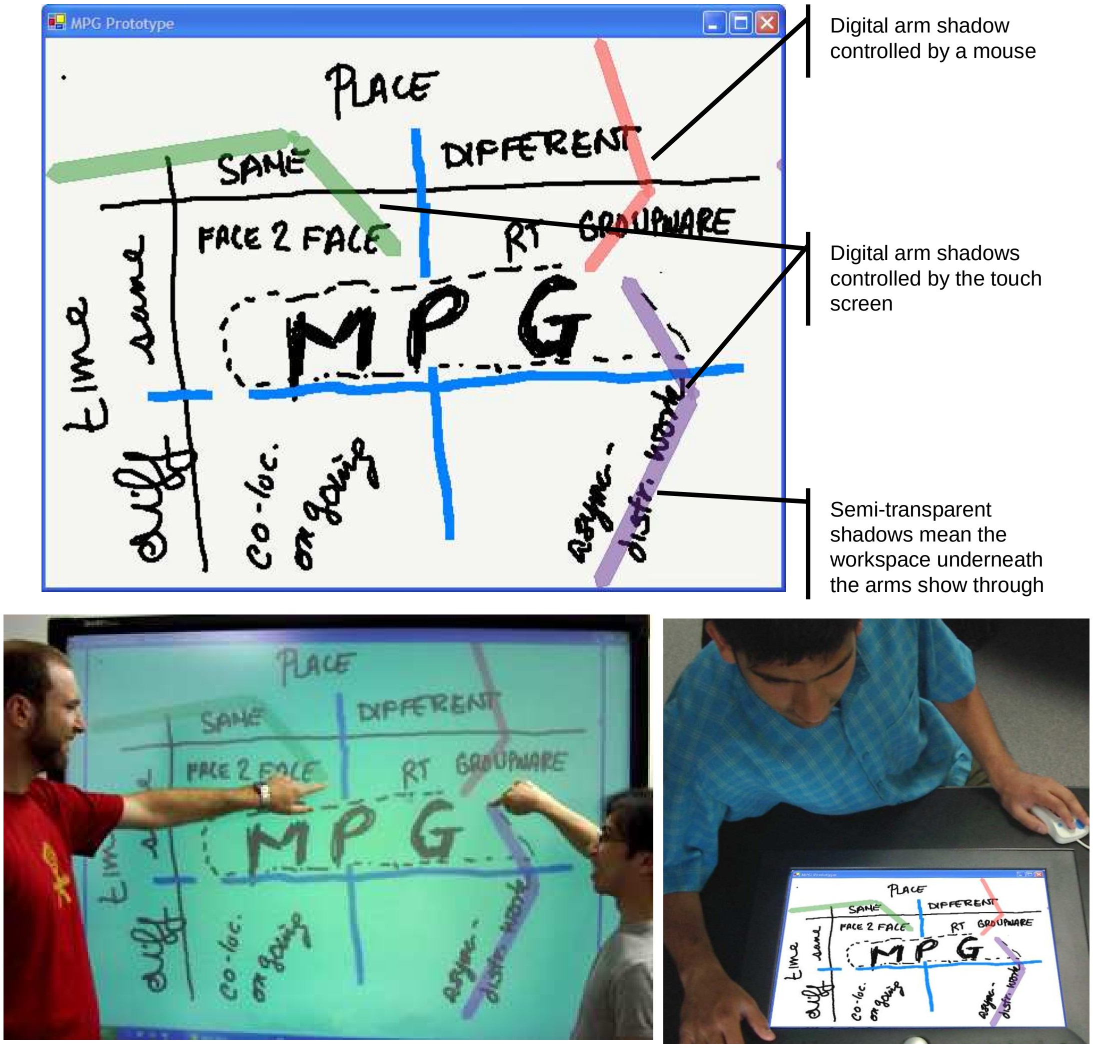
4.2 DigitalArms

I prototyped an embodiment system I call DigitalArms (Figure 4.1). The system detects the participant's physical presence and seating orientation around the workspace, and tracks the participant's mouse movements in the workspace. Participants are represented by semi-translucent "arms" that appear to be shadows, which means the workspace underneath is visible.

In this section, I describe an approach to designing and building DigitalArms, which is to mimic visible properties exhibited by corporeal bodies in physical workspaces. I focus on the properties presented earlier in this chapter, and show how the arms were designed to mimic physical, corporeal arms.

4.2.1 Detecting user presence

One of the key problems I aimed to address with DigitalArms was the problem of presence. Typically, distributed groupware presents an embodiment (i.e. a telepointer) for each enumerated input device when a network connection is present; however, this approach is inappropriate for MPG, because there may not actually be that many participants interacting with the workspace. Specifying the number of participants when starting the application is too restrictive—participants in the workspace may arrive late, or leave early, which would require restarting the application. Instead, I designed two implicit mechanisms to detect user presence. First, the system recognises occupied seats around the



Digital arm shadow controlled by a mouse

Digital arm shadows controlled by the touch screen

Semi-transparent shadows mean the workspace underneath the arms show through

Figure 4.1. Semi-transparent digital arm shadows provide cartoon-like arms for remote collaborators.

table by an embedded light sensor in the seat. When the light sensor no longer detects light, the system understands that someone has sat down (sensors are implemented using Phidgets: Greenberg & Fitchett, 2001). Of course, this solution requires fixed seating—since a seat is implicitly bound to some input device, moving seats around the table would require system recalibration.

My second mechanism for detecting user presence monitors the mice movements, where each mouse is associated with a particular seat. The system detects the absence of the user through an inactivity timeout of three minutes. By combining information about

	Last mouse activity			
	Within 30s	**Within 1 min**	**Within 2 min**	**Within 3 min**
Seat occupied	Present	Present	Present	Present
Seat free	Present	Likely present	Possibly present	Absent

Table 4.1. A model of presence based on the state of two inputs: time since last mouse activity, and seat occupancy.

the seating of participants and mouse activity, we can develop a reasonable model of presence, summarised in Table 4.1. The system takes the seat occupancy as the primary indicator of presence: if the seat is occupied, the system assumes the participant is engaged in the activity. The mouse activity timeout takes lower precedence, but is important if a seat is not occupied (which occurs when a participant uses a mouse without sitting down). The more time that passes without activity, the less certain the system can be about the user's presence.

Clearly, this model of presence is fairly simplistic and prone to error. For instance, do we present embodiments for lurkers—those who watch, but do not actively participate in the collaboration on the workspace? If not, their presence may have an effect on co-located users, but not on remote users. Similarly, the system may incorrectly infer the absence of a participant even though the participant is simply standing and engaged in discussion. Furthermore, a fairly large body of literature conceives of presence as a deeper notion with many facets (for a review, see Lombard & Ditton, 2001). In spite of the simplicity of this model, it is a reasonable first attempt to model presence beyond the common "network connection is presence" model.

With these methods of detecting presence in hand, I now discuss the DigitalArms as a method for representing and presenting presence information.

4.2.2 DigitalArms as indicators of social presence

For inspiration, I turned to VideoWhiteboard (Tang & Minneman, 1991b), the video-based tool that provides a large shared drawing area between two sites I introduced in Chapter 3. In VideoWhiteboard, video cameras behind semi-translucent drawing surfaces capture all activities on and near each surface, including not only the marks made on the surface with a felt pen, but the shadow of the users' bodies (usually hands and arms) as they move atop it. The video from both sides are then fused, creating a composited image, as illustrated in Figure 3.6. Because the cameras capture images through the semi-translucent surface, each participant's arm gracefully appears as a shadow on the workspace as he moves toward the workspace, and disappears as he moves away from it. These arms are not only visually large, they are also socially natural indicators of presence. While extremely effective, VideoWhiteboard has technical limitations as mentioned in Chapter 3. First, it does not scale well since each additional person adds an additional overlayed video stream, meaning the image degrades fairly rapidly. Second, it has high setup and equipment costs. Finally, people cannot edit each others' marks, meaning it is limited as a vehicle for collaboration.

DigitalArms for remote collaborators incorporate some of the properties of presence found in VideoWhiteboard (as illustrated in Figure 4.1). Using real arms working over tables as my model (such as Figure 2.6), each arm shadow maintains a 135° articulation, and natural forearm/upper-arm and width/length proportions. The "shoulder" point of the arm is attached to one of the sides of the table, and the "hand" point is bound to the mouse cursor location. The arms themselves are semi-transparent, allowing objects on the underlying workspace to show through. Finally, their visibility, as described above, is based on the detection of the presence of an individual, be it through light sensors or activity. When presence is uncertain, the transparency of the DigitalArm slowly increases until it disappears altogether.

DigitalArms are packaged as an independent software component (i.e. a widget) that can be incorporated into MPG applications. Through a simple programmatic interface, the programmer can bind the hand of the DigitalArms to telepointer locations, and the shoulder point to several “seating” positions around the display.

I then replaced MPGSketch’s telepointers with arm shadows to represent participants. Figure 4.1 gives an example, with two people at the East and West sides of a large display, and one person at the North side of a table. To show presence and absence (as in Table 4.1), the DigitalArm appears only when a user’s presence is detected, and disappears when the user leaves. The software embodiment thus has a property of a real-life embodiment: the embodiment is only present when the person is physically present and active over the surface. In contrast to most other groupware systems, this embodiment system differentiates between a person’s presence at the terminal as opposed to the software client’s connection to the server.

To summarize, DigitalArms reproduce several key attributes of real-life embodiments (as in Figure 2.6) beyond those offered by standard telepointers.

1. ***Indicates virtual seating position***. DigitalArms appear from a side of the application window frame much as a person’s corporeal arms appear from a person’s seating position, thereby “grounding” the virtual arms to an imagined virtual body.;

2. ***Conveys person-specific orientation.*** Each arm has a different orientation, fostering the impression that each user has a distinct view of the display.

3. ***Increased awareness of actions.*** A participant’s actions are far more visible to others when compared to telepointers. First, our transluscent shadows partially obscure the workspace underneath the arms, mimicking how real arms obscure parts of the workspace as they work (Figure 2.6). Second, digital arm shadows are large (about an order of magnitude larger than telepointers), thereby commanding more attention.

4. ***Transmits identity.*** Current customizable arm parameters include colour and proportion. Adding video portraits can increase the ability for collaborators to identify one another at the cost of additional screen space.

These properties of DigitalArms, taken together, are virtualizations of real-life properties found in corporeal arms above and beyond those offered by standard telepointers. And, while the DigitalArms system was little more than a simple prototype, it demonstrated the promise of reproducing visible aspects of an individual's presence in the workspace, much like video-based embodiments (as discussed in Chapter 3). Given the success of DigitalArms (which are essentially telepointer-based embodiments), it made sense to recreate the idea using video. A video-based technique following in the spirit of DigitalArms would be able to reproduce the success with the benefit of increased video fidelity. Thus, I began work on VideoArms.

4.3 VideoArms: an embodiment system for MPG

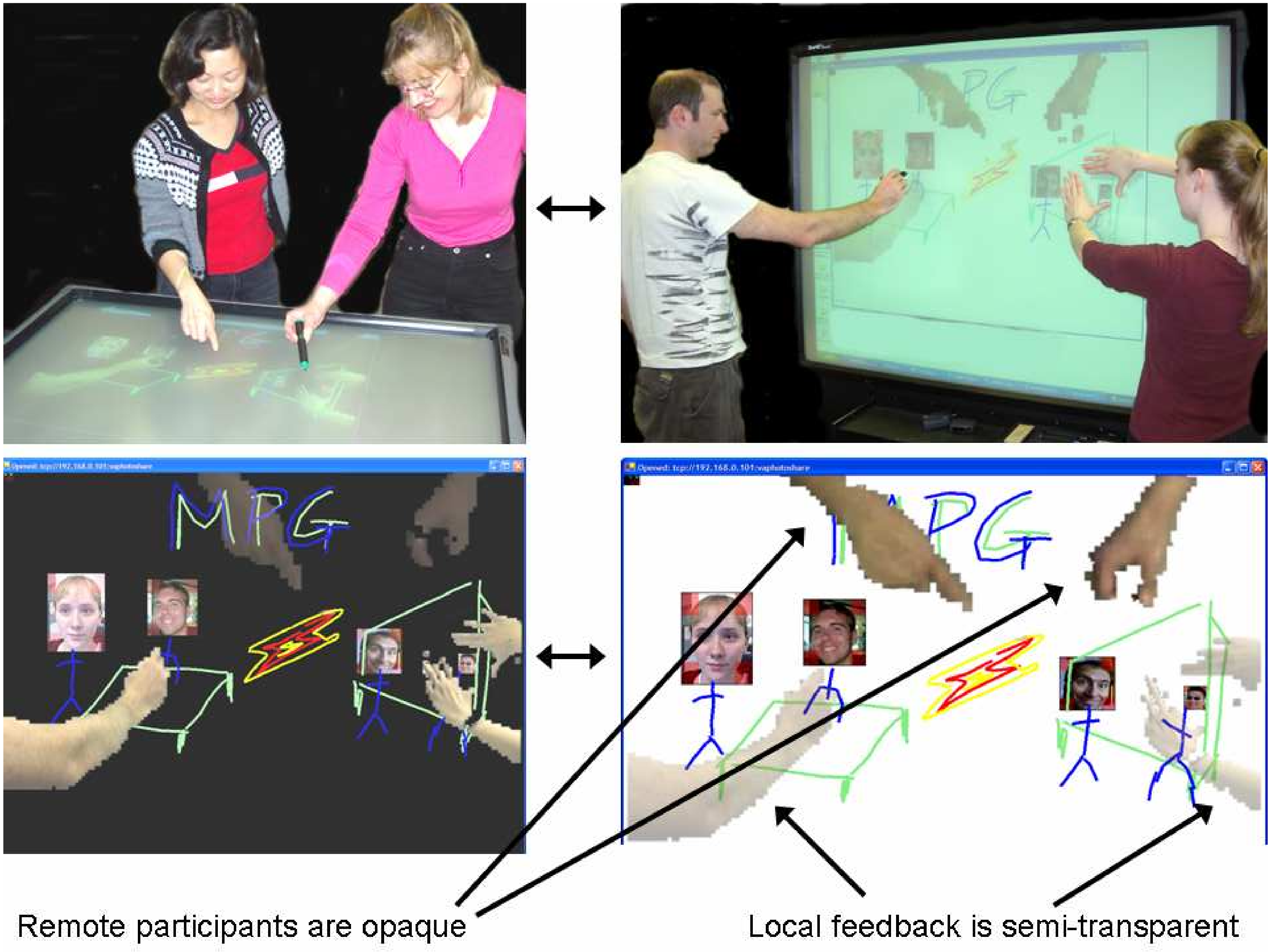
VideoArms captures collaborators arms as they work over the workspace using a video camera, and redraws the arms at the remote location, producing an even richer effect than the DigitalArms system. Figure 4.2 illustrates a snapshot of a sample session of VideoArms, and I will use these images to explain how VideoArms work. The top images show two connected groups of collaborators. Each group works over a touch-sensitive surface—the left is a front-projected touch-sensitive horizontal DViT, while the right is a rear-projected vertical SmartBoard. The surface displays a custom MPG application that lets people sketch and manipulate images, while displaying video embodiments. The bottom set of images are screen grabs that reproduce what these groups see on the shared display. Not shown are cameras situated in front of the displays (Figure 4.3).

The figure illustrates what participants can see and do with the VideoArms embodiment system in this MPG application. First, collocated people can see their own arms as local feedback. These are rendered as semi-transparent, shadow-like images, providing feedback of what others can see while minimizing interference (Zanella & Greenberg, 2001). Second, each group sees the solid arms of the remote participants in reasonable 2½-dimensional fidelity, meaning that while the images are not truly 3-dimensional, the system captures and reproduces colour-based depth-cues. Third, this is an MPG setting where all participants can simultaneously gesture to the full, expressive extent of arms and hands. This system neither dictates nor implies any sort of turn-taking mechanism, and captures workspace and conversational gestures extremely richly. Fourth, both physical and video arms are synchronized to work with the underlying groupware application, where gestures and actions all appear in the correct location[1]. Fifth, arms preserve the physical body positioning relative to the workspace. For example, because the people at the table display are standing at the back side of the image, their arms appear on the vertical display as coming from the top. Finally, participants are not tethered to any particular place in the workspace: using touch and pens to interact with the groupware application, participants are free to physically move around the workspace as they see fit.

In Chapter 3, I derived four design implications for MPG embodiments. Briefly, the implications are as follows: (1) support visible embodiments for local and remote participants, (2) support consequential communication by using direct input mechanisms, (3) support both conversational and workspace gestures, and (4) support gestures in the context of the workspace. From a collaborative standpoint, VideoArms satisfies these requirements.

1. Local participants know what remote people see because feedthrough is shown as feedback as semi-transparent.
2. Consequential communication of actions is supported well because the body is the input device on the touch sensitive surface. Other collaborators can easily predict, understand and interpret another's actions in the workspace as one reaches towards artefacts and begins actions.

[1] VideoArms simply reproduces a video-captured image of the workspace. In principle, it can therefore support an infinite number of non-overlapping arms. While my goal was to develop a true MPG application with VideoArms, technical limitations imposed by the input devices (the actual SMARTBoards) meant that my final system only supported two simultaneous touches on one display; the other display could only support a single touch.



Remote participants are opaque

Local feedback is semi-transparent

Figure 4.2. VideoArms in action, showing two groups of two people working over two connected MPG displays (top) and a screen grab of what each side sees (bottom). Local and remote video arms are in all scenes, but local feedback is more transparent.

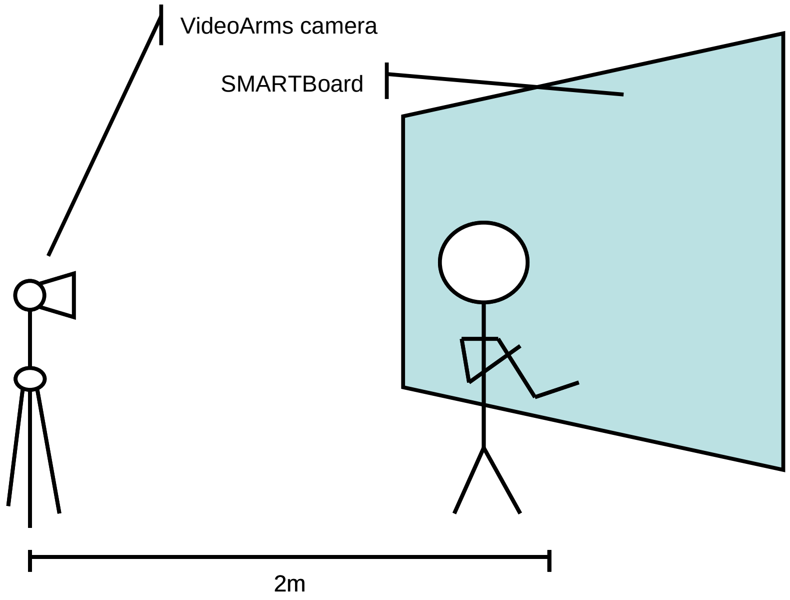


Figure 4.3. The setup of the VideoArms camera with respect to the touch sensitive SMARTBoard.

3. Rich gestures (coupled with conversation and artifact manipulation) are also supported well because the remote arms are displayed in rich 2½ dimensional fidelity and a reasonable framerate (~12 fps). While clearly not ideal, practical experience with the prototype showed that 12 fps was reasonable enough to interpret gestures. Finally, task-related gestures are easily interpreted because they are placed in the context of the workspace.

4. Collocated participants can use and interpret natural body language of their physical bodies as they collaborate. Because collaborators are not tethered to input devices, their actions are direct and in the workspace context; thus, an individual's physical body *is* the embodiment.

4.3.1 Implementation

VideoArms uses inexpensive web cameras positioned approximately two meters in front of the display (Figure 4.3). The software extracts the arms (and other bare-skinned body parts) of collaborators as they work directly over the displayed groupware application. It transmits these images to the remote workstation, where they are further processed to

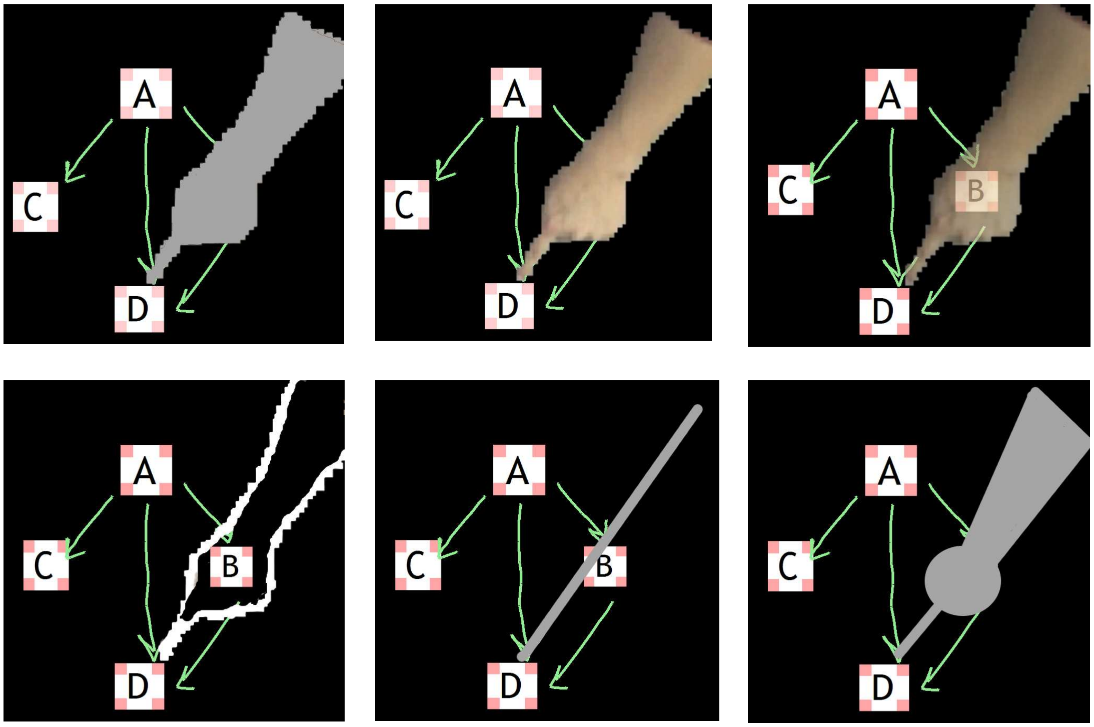


Figure 4.4. Various manipulations of a VideoArm. The stick and ball-and-stick manipulations are possible, but not implemented.

appear as an overlay atop the digital workspace. To provide local feedback, it overlays a local person’s video on the work surface.

Frames captured by the camera are processed, transmitted and displayed in a four step process.

The first step finds the regions in the video frame that match skin color. This step is fairly expensive computationally, and is described more fully in the next subsection. Morphological opening, a standard computer vision technique, is then applied to the skin mask to remove image noise while still preserving the shape and size of larger objects. This process produces a silhouette image of the collaborator’s arms similar to what is seen in Figure 4.4 (top left) and to the shadow-like embodiment found in other systems (e.g. Tang & Minneman, 1991b).

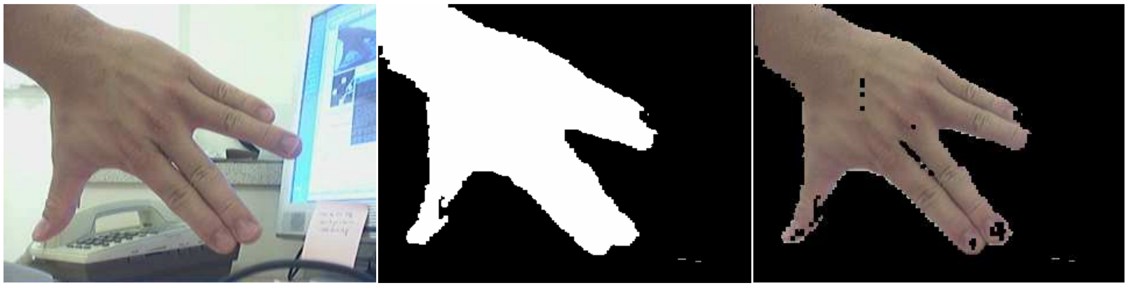


Figure 4.5. The image on left is colour-segmented to find the skin-colour pixels (middle image). The two images are then combined to produce the VideoArms image on the right.

The second step produces a full-colour image of the real arms using the original image and the silhouette image from the previous step (Figure 4.5). It does this simply by overlaying the silhouette with pixels from the original image: black pixels of the silhouette are copied onto the original image. The result is that only the skin-coloured pixels are preserved.

The third step transmits this image to listening clients via IP multicasting (clients include both the remote and local display). IP multicasting is used to reduce the amount of data on the network, and its use of UDP packets ensures quick delivery. Of course, other networking techniques could be used but care has to be taken to preserve performance.

The fourth step uses standard GUI techniques to draw all received images on top of the groupware work surface, which creates a composite of local and remote arms.

Because the VideoArms are completely digital images, they can be rendered in many ways. For example, several arms in Figure 4.2 and the arm in Figure 4.4 (top right) are rendered semi-transparently. Other possible techniques include outline, vector, and stylized arm representations (Figure 4.4, bottom), and a means to change the size of the arm (Roussel, 2001). As well, unlike analog video systems, the digital nature of our VideoArms means that it can be applied to any kind of groupware system. Finally, as mentioned earlier in Chapter 3, analog video systems suffer from the drawback of degraded image quality when multiple video signals are composited. The digital nature of VideoArms does not suffer from this drawback since image noise can be digitally removed.

VideoArms is built using Python, the .NET Framework, PyIPP (a set of Python wrappers for the Intel Performance Primitives library), the Python Imaging Library, and the Python numarray open source libraries. Several inexpensive cameras were used, ranging from an Intel CS430, a Logitech QuickCam Pro 4000, and a Winnov Videum camera. To maximize performance, I use one computer to process and transmit the captured video (ideally this could be done by a special purpose hardware board), and another to display the VideoArms and run the groupware application. On a Celeron 2.4GHz PC video frames are processed at 320×240 resolution at 25 frames per second, which is overlaid across a high resolution 640×480 groupware workspace. This resolution is reasonable for interpreting consequential communication and gestures. It also improves upon LIDS, which works over a 640×480 workspace and a 176×144 video image on a 1GHz machine (Apperley et al., 2003).

4.3.2 Color segmentation

The primary computational cost of VideoArms is the first step described in section 4.3.1, which determines where skin is located in an image. My primary approach to this problem has been to use colour-based image segmentation techniques. To be clear, there exist other methods that involve using external knowledge about the image. For example, Starner & Pentland (1995) use hidden Markov models while Takahashi & Kishino (1991) rely on data glove-like input devices. Another approach is to use expensive infrared or heat-detection cameras, as in (Miwa & Ishibiki, 2004); however, for the purposes of my prototype, the colour-segmentation approach was sufficient to capture arms to test my ideas.

I have tried two approaches for colour segmentation. The first uses the fact that in HSV (hue, saturation, value) colourspace, skin tones are contained in a small, fairly well-defined space, regardless of race (Boyle, 2001). Images are converted from RGB (red, green, blue) to HSV (hue, saturation, value) colourspace, and a brute force matching algorithm determines which pixels in the image correspond to skin tones, thereby creating a skin mask. The problem with this approach is that setup for the system is time-consuming (in the order of 20 minutes), and is required each time a new camera is used. A corpus of

10 images must be hand-marked for “skin colour” pixels and then processed by the system before use.

Another approach I tried involved a simpler calibration step, and uses a statistical quantity known as the Mahalanobis distance to determine the likelihood of a given part of an image as being skin. With this approach, samples of skin are taken (this is done in a calibration step) by taking a picture, and specifying 10 points on the image, whose colour (R, G, B) values are read. Given these values, a mean vector, and covariance matrix are calculated. The mean vector is the average R,G,B values: analogous to the mean, or the centre of a sample. The covariance matrix is the multivariate analogue to the variance measure: the covariances between the variables sit along the non-diagonal positions, and variances of the variables are placed along the diagonal. The covariance matrix is a measure of the extent to which the values vary with one another. The Mahalanobis distance is analogous to the Euclidean distance (except that it takes into account covariances). It is calculated for a new (R,G,B) value against the mean vector and covariance matrix, and is a measure of “typicality” of the value given sample. If it is typical (an arbitrary cut-off), then the pixel is judged to be skin; otherwise, it is judged to not be skin.

The Mahalanobis distance approach can be used interchangeably with the brute-force HSV-matching approach described earlier. Both approaches produce similar results, and both are CPU intensive; however, the Mahalanobis distance approach is easier to calibrate. I therefore chose the Mahalanobis distance approach for my final prototype.

4.3.3 Performance considerations

In designing VideoArms, I considered carefully the trade-offs between image fidelity versus the smoothness (frame rate). Consequential communication is timely (Segal, 1995), so VideoArms’ images needed to be capable of being transmitted as rapidly as possible. A higher fidelity image could convey more information to collaborators, but at the cost of increased transmission time. Similarly, gestures need to be conveyed as smoothly as possible to ease comprehension (Gutwin & Penner, 2002); thus, smaller images are

Image resolution	Colour segmentation	Color segmentation with morphological opening
160×120	33.34s (29.99 fps)	33.34s (29.99 fps)
320×240	85.93s (11.63 fps)	91.96s (10.87 fps)
640×480	375.78s (2.66 fps)	410.65s (2.43 fps)

Table 4.2. Processing times for 1000 frames under different processing requirements at various image resolutions.

preferred as they can be transmitted more rapidly. However, small images are low fidelity, and prone to noise. Thus, I needed to consider different image sizes (320×240 and 640×480 were reasonable), different approaches to clean up the image (morphological opening, small-object noise removal), and finally different mechanisms to transmit the image to clients (client-server, direct network connections, and multicast). Clearly, the ideal would be to have high resolution images transmitted at high frame rates with low image noise; however, it is clearly impossible. Thus, in making technical choices, I preferred high frame rate and clean images over high fidelity images since the latter requirement had the most dramatic effect on performance.

Transmitting larger images produces many desirable benefits for VideoArms. The problem is that larger images take longer to transmit and longer to process both at the originating machine, and at the destination machine. For instance, doubling the image resolution from 320×240 to 640×480 increases the work four-fold since there are four times as many pixels (76800 pixels *vs.* 307200 pixels). Table 4.2 shows some sample data produced on a Pentium 3 550 Mhz machine that demonstrates the increased processing time required for images of various resolutions. This increased processing time is matched by an increased amount of time required by client machines to draw the images, and finally increased network transmission time (though the data is difficult to produce since network transmission times are generally quite low). In keeping with the preference for high frame

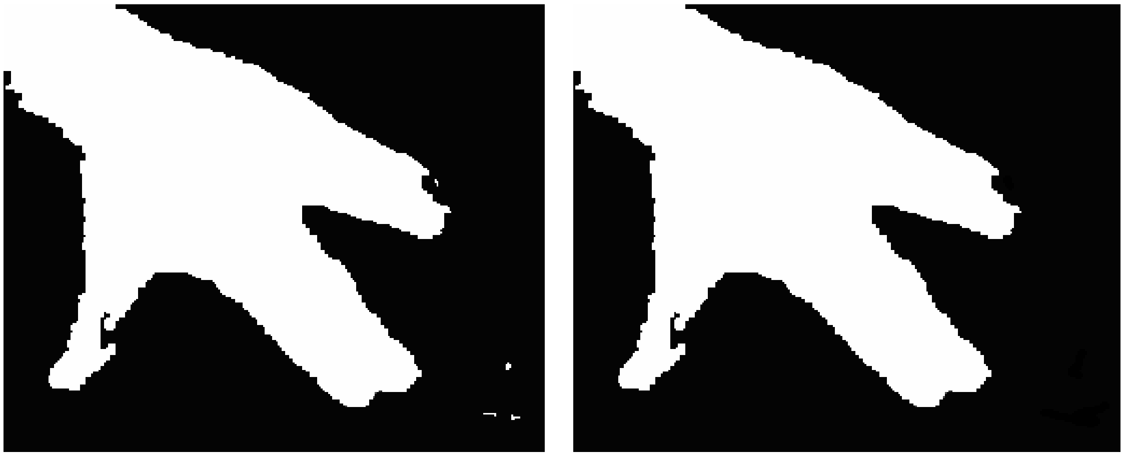


Figure 4.6. The small region removal algorithm removes errors in the left image (seen in the bottom right of the image) from the final image on the right. Note that this example is contrived to show the effect. Real-world images generally have a number of false-positives.

rates, I chose a 320×240 resolution for my final prototypes since my algorithms for image processing were too rudimentary to support higher resolutions at the desired frame rates.

The images produced by the colour segmentation algorithms described in section 4.3.2 produce "noisy" images, meaning that there are occasional errors (e.g. Figure 4.6). There are two types of such errors: false positives (where the algorithm falsely identifies a pixel in the image as skin-colour), and false negatives (where the algorithm fails to identify a skin pixel as a skin pixel). The result of these errors is that the image contains confusing artefacts. To remove these errors, I tried two different techniques: small object removal, and morphological opening. Small object removal is an image processing technique that takes an image and deletes arbitrarily defined "small" regions from the image. The process is designed to remove false positives from the image, thereby cleaning up the image by removing artefacts (Figure 4.6). The drawback of this technique is that process is CPU-intensive, increasing the amount of time it takes to process each frame. The second technique, morphological opening, is a technique that removes false negatives in the image. It does so by "dilating" the image followed by "eroding" the image (Figure 4.7). The dilation and erosion processes specifically concern pixels that are the border between black and white regions of the image. Dilation simply enlarges the conceptual "size" of these

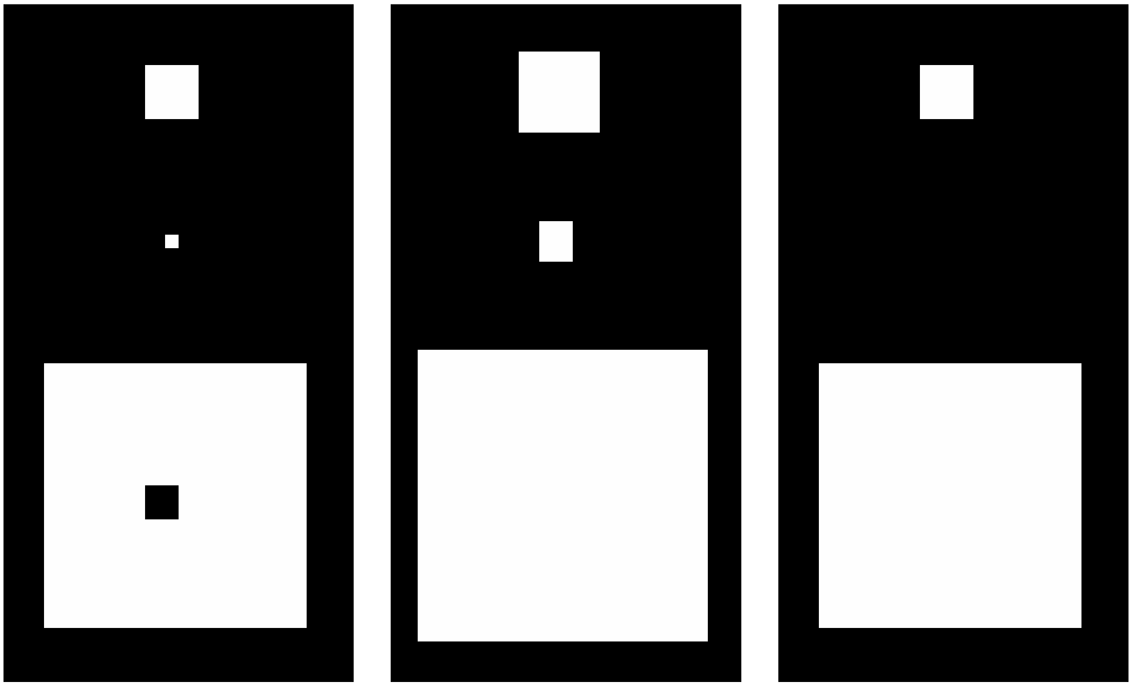


Figure 4.7. Morphologically opening an image is the result of alternately dilating, then eroding the image. The result of dilating the image on the left can be seen in the centre image, and the final result of eroding the centre image can be seen in the right image. The image on the right is therefore the result of a morphological opening process applied to the image on the left. Notice that the small black box disappears after the dilation process while the small white box disappears after the erosion process while larger objects are left unaffected.

pixels (similar to how a pupil, when dilating, grows larger in place). Conversely, erosion shrinks the conceptual "effect" of these pixels. This process effectively removes false negatives from the image. In my tests, it also had the added benefit of removing false positives from the image, and took the same amount of time as the small region removal. I decided ultimately to use morphological opening (over not using any image cleaning algorithms) since noise in the image had the potential to confuse users of VideoArms—especially given my desire to have fine-grained, high fidelity images and already having compromised on the resolution of the images.

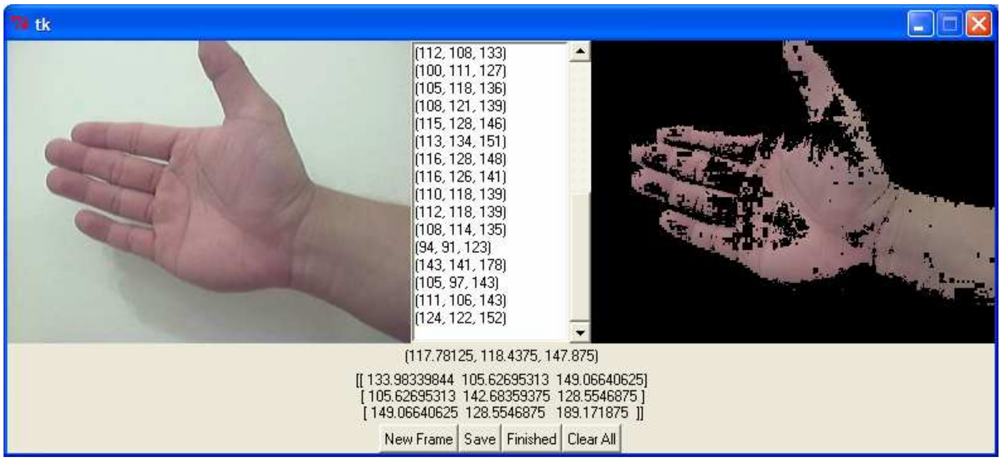
Finally, I considered several different mechanisms for image transfer between the machines. A client-server approach (using the Collabrary) seemed architecturally appropriate since it could leverage the built-in networking facilities. However, the Collabrary was built intending to support high throughput of only small data items; images are at least an order magnitude larger than the intended data size. As a result, the Collabrary did not adequately support the sending of image data at the desired framerate. A second approach would be to set up dedicated TCP network connections between clients. While such an approach seems reasonably appropriate, it does not scale well. Adding more listeners of VideoArms data increases the amount of data traversing the network dramatically (since each would require its own TCP connection). Furthermore, TCP packets inherently travel slower across a network than UDP packets. Thus, I chose a network transmission methodology called IP multicasting, where instead of sending data to destination machines on the network, a machine simply sends data to a multicast address (akin to a television channel). Interested parties subscribe to the multicast address, and they can receive the data. IP multicasting has the benefits of using high speed UDP packets, and does not need to replicate the data sent across the network with an increased number of listening clients.

The final prototype of VideoArms uses 320×240 images, performs morphological opening, and sends the images using IP multicasting. While further optimizations are possible, my primary intention was to develop a system suitable to test my ideas and not to produce a production-level implementation. Thus, 12 fps and the reasonably crisp images that I get with this set of technical decisions is sound.

4.3.4 Calibration

The current VideoArms implementation requires two calibration steps on a per-location/per-camera basis.

First, each time a new camera is used, the system must be calibrated to understand “skin tones” from the camera. Because webcams are generally of fairly low quality, their picture quality, colour range and image sharpness differ drastically from model to model. As a result, colour components (R,G,B, and their counterparts H,S,V) that register on one camera as skin may not register as skin on another and vice-versa. To calibrate the system for the brute-force HSV approach, I use a corpus of ten images taken with the camera to determine appropriate values for skin tones for that particular camera. For the Mahalanobis



```
(112, 108, 133)
(100, 111, 127)
(105, 118, 136)
(108, 121, 139)
(115, 128, 146)
(113, 134, 151)
(116, 128, 148)
(116, 126, 141)
(110, 118, 139)
(112, 118, 139)
(108, 114, 135)
(94, 91, 123)
(143, 141, 178)
(105, 97, 143)
(111, 106, 143)
(124, 122, 152)

(117.78125, 118.4375, 147.875)

[[ 133.98339844  105.62695313  149.06640625]
 [ 105.62695313  142.68359375  128.5546875 ]
 [ 149.06640625  128.5546875   189.171875  ]]
```

Figure 4.8. This wizard allows me to select examplar pixels on the left to train the system what colours are “skin” colours. The list in the middle shows selected (R,G,B) values along with the centroid and covariance matrix. The right image shows how the image on the left would be segmented with the current set of variables.

distance approach, I need only one image from the camera, and simply specify 10 points on the image that correspond to skin (Figure 4.8).

Second, to correct for imperfect camera-screen alignment, a short five second calibration sequence is run when VideoArms is started. The problem arises because the camera is rarely positioned such that the groupware application perfectly fills the camera frame. A simple calibration wizard shows the camera frame, and asks the user to select three corners of the groupware application. From here, the wizard determines how much of the camera image to crop, and how the image needs to be transformed so that the composited image displays arms accurately.

4.3.5 Implementation nuances

As I developed VideoArms, I identified two difficulties that were solved via workarounds. I include my solutions for those wishing to replicate and perhaps improve my implementation.

First, VideoArms performs image segmentation to determine where “skin” is in the image. This approach works well when the skin/hands are being used over a rear-projected display, plasma or CRT display. However, in front-projection systems, detecting skin is more difficult: people’s hands are interposed between a projector and the physical surface and the bright light of the projector shining on people’s hands washes out their skin tones. To reduce skin discoloration, I limited the color palette of the front-projected groupware workspace to dark tones (e.g., black, brown and evergreen). A better solution could be to predict and detect the discoloration on the skin given the particular pixel colors being projected; however, such an algorithm is not well understood and is likely computationally expensive. Finally, a better implementation could do away with colour segmentation altogether. For instance, the use of thermal-imaging (Miwa & Ishibiki, 2004) or infrared cameras (Ishii, Nakanishi, Koike, Oka & Sato, 2004) can produce very clean results at the cost of expensive equipment.

With high quality video cameras and bright projection displays, the camera can capture not only people’s physical bodies, but also the VideoArms projected on the workspace. This can result in visual feedback loops if the algorithm perceives the projected remote VideoArm images as skin. To solve this problem, I paint the images of remote arms slightly off-colour so that they are not captured by the system. This seems to work well in practice. LIDS also report this problem, but they use a more complex image-processing technique to remove shadows after they are captured (Apperley et al., 2003).

As seen in Figure 4.2, my VideoArms are not perfect. While certainly useful, they are somewhat jaggy and noisy. They also appear at roughly 12 frames per second rather than the 30 fps recommended in cinematography. This is because I am primarily interested in contributing to CSCW design research *vs.* computer vision research; I use only elementary and well-known image processing techniques in my prototype. Undoubtedly, computer vision researchers could improve on my method of extracting arms from the scene while minimizing processor demands.

4.4 Summary

In this chapter, I described the evolution, design and implementation of the VideoArms embodiment system for MPG (Goal 3). In going through the evolution of the system, I discussed an early system called DigitalArms, which motivated the VideoArms work. I also detailed how the VideoArms design theoretically fulfills the requirements for MPG embodiments that I derived in Chapter 3.

Yet, how does this system bear out in practice? In the next chapter, I describe a full observational study of the VideoArms embodiment system. In it, I will evaluate VideoArms' effectiveness in fulfilling the requirements I specified in Chapter 3, thereby mitigating presence disparity (Goal 3).

Chapter 5. Evaluating VideoArms

In Chapter 3, I proposed four theoretical design requirements for mixed presence groupware embodiments to mitigate presence disparity, where they should support: (1) local feedback, (2) consequential communication, (3) rich, timely gestures, and (4) gestures within the context of the workspace. In Chapter 4, I used these requirements to motivate the design of VideoArms as an embodiment technique. In this chapter, I evaluate and critique VideoArms' ability to address presence disparity via an observational study (Goal 3). We will see that the VideoArm concept is well suited to the collaborative workspace by virtue of its exceptional support for gestures. However, we will also see that presence disparity still persists.

I begin by recapping the key design differences between VideoArms and telepointer-like embodiments. I then describe two tasks I created to explore these differences: the highly constrained *elastics and nails* task, and an open-ended design task that allowed participants to freely use the drawing surface as they saw fit. In the following sections, I describe the methodology, present key results, and discuss the findings. In particular, I will argue that VideoArms outperforms telepointer-like embodiments as a mechanism to engage remote participants. However, it is not yet ready for prime time: its imperfect implementation hinders its acceptability as an embodiment technique.

5.1 VideoArms features

In principle, embodiment techniques such as telepointers (as in Figure 3.1) or DigitalArms (as in Figure 4.1) fulfill the design requirements from Chapter 3. Telepointers provide local feedback, approximate certain pointing and motion-based gestures, are placed in the workspace, and are visible (to some extent) while the owner is working. Yet telepointers

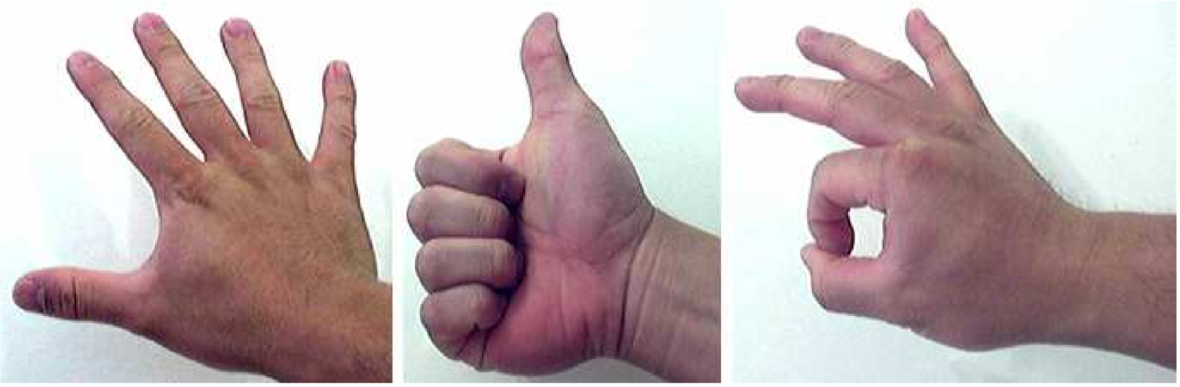


Figure 5.1. Common real-life gestures that were observed in testing of VideoArms.

only go so far because they present only X, Y location information about the user. DigitalArms enriches telepointers by adding a static “shoulder” representing seating position. DigitalArms also makes actions and movements more salient through size and animation without adding new information. While far better than no embodiment, both approaches confine the user’s actions and impact on the workspace due to their low information richness and fidelity.

In contrast, VideoArms give users the capacity to use their arms and hands as an input device. Thus participants should be more capable of fluidly and directly expressing themselves through a rich variety of gestures, be it with their hands and or with their whole arms. The richness of video should also help convey consequential communication to remote participants. Thus, VideoArms better implements the four theoretical design principles, providing greater information richness, and the opportunity for novel interaction and communication strategies (e.g. using both hands in the workspace, or using natural hand gestures as in Figure 5.1).

I previously said that presence disparity is characterised by a markedly differential engagement between local and remote participants. If an embodiment technique mitigates presence disparity, we should see increased engagement between remote participants. I hypothesize that VideoArms’ increased fidelity and communicative capacity, through its richer support for gestures and consequential communication, should increase the overall level of engagement between remote participants. To test this hypothesis, I ran an observational study comparing VideoArms to DigitalArms, where I considered the latter as a benchmark similar in utility to telepointers.

5.2 An observational study

I designed and ran an observational study to evaluate VideoArms' support for the four design features specified in Chapter 3, and to understand whether these collectively mitigate presence disparity. I evaluated VideoArms in three separate stages, each with a slightly modified protocol to investigate a different set of questions. I begin by articulating these questions, followed by a description of the participants, the materials and tasks used in the evaluation. I then discuss the experimental procedure, the data collected and finally justify the tasks in the evaluation. The results and discussion are presented in subsequent sections.

5.2.1 Questions of interest in a three-staged approach

VideoArms' approach to mitigating presence disparity is to present remote collaborators through the richness and fidelity of video. Theoretically, this approach allows gestures, workspace activities and consequential communication to be natural, easily conveyed and interpreted. Thus, in this study, I am interested primarily in uncovering the incidence and variety of gestures, the occurrence of consequential communication, and the level of engagement between remote participants.

5.2.1.1 Stage 1: Establishing basic utility

I first needed to establish that individual participants would make use of VideoArms' support for gestures and consequential communication across a distributed display. In Stage 1, I observed if participants in groups of two (one per display) without a voice link were able to overcome the lack of a voice channel by using their embodiments. Specific questions at this stage in the evaluation included the following.

- Do participants gesture within the context of the workspace?

- How natural are these gestures?
- Are there obvious instances of consequential communication?
- Do participants make use of local feedback?

5.2.1.2 Stage 2: Evaluating utility in mitigating presence disparity

Next, I evaluated VideoArms as a method for mitigating presence disparity. Specifically, in Stage 2, I observed whether groups of four (two per side) comprising collocated and remote participants, made use of VideoArms to engage remote participants in the presence of a voice link.

- Do participants use gestures even though there is a voice link?
- How are gestures used in the workspace? For whom are they intended? Are gestures repeated or modified for remote participants?
- Does consequential communication occur and is it used across the link in spite of the presence of a voice link and a collocated participant?
- How does correction (a common occurrence related to consequential communication) occur? Do participants generally only come to the aid of collocated participants, or do they also aid remote participants?

5.2.1.3 Stage 3: Understanding the tradeoff between local feedback and high frame rate

To foreshadow, I observed that participants' gestures in Stage 2 were unnaturally slow. One possible explanation for this behaviour is that participants deliberately slowed down the pace of their gestures to compensate for the low framerate (caused by the composition of local feedback on the remote image). To verify this hypothesis, I used the same procedure as in Stage 2, except to increase the frame rate of remote participants, I removed local feedback. Note that I am not discounting the utility of local feedback. Instead, I am interested in observing whether increased remote frame rate would facilitate more naturally paced gestures. This third stage was run with only one group, as it was not the primary focus of my evaluation. However, I include it here for completeness.

- Give higher frame rates, are gestures conveyed at a more natural pace? If not, are they at least completed more rapidly?
- Are task completion times more rapid?
- Are participants hindered by the lack of local feedback?

To summarise, I divided my evaluation of VideoArms into three separate stages. In the first stage, I establish the baseline utility of VideoArms. In the second stage, I examine whether VideoArms mitigates presence disparity in a constrained and open-ended task. Finally, in the third stage, I briefly examine the effects of increasing the frame rate of remote participants. Together, these three stages of the evaluation provide a reasonably complete picture of the utility of VideoArms in mitigating presence disparity.

5.2.2 Participants

A total of 22 paid participants were recruited to evaluate VideoArms. Participants were recruited via an electronic message of the day seen by users of the main UNIX server for the Department of Computer Science, and by an email sent to that department's graduate students. Participants, 12 females and 10 males, ranged from 18-29 years of age. All were computer-proficient, using computers daily, and 18 of 22 participants were computer science majors or graduate students (the remaining participants were students from other faculties). Most participants (17 of 22) used an instant messaging application (e.g. MSN Messenger) on a regular basis, so were familiar with distributed groupware. However, only six of the participants had previously used computer-whiteboarding tools. Finally, only three had formal HCI education, all being currently involved in an HCI class (two at the undergraduate level, one at the graduate level).

Participants were recruited as groups, so each participant already knew his or her group members well. Six of these participants were pairs, and participated in Stage 1 of the evaluation (3 groups of 2). 12 participants were in groups of four for Stage 2 of the evaluation (3 groups of 4), while the remaining group of four participated in Stage 3 of the evaluation (1 group of 4). Thus, a total of 22 participants or 7 groups were observed using VideoArms.

5.2.3 Materials

Four Celeron 2.4Ghz machines, each with 256 MB RAM and connected on a 10Mbps hub on a private network, were used in the evaluation. Two of these were connected to Logitech QuickCam Pro 4000 cameras, and served as VideoArms capture machines, responsible for capturing and sending out images of collaborators as they worked. The remaining two machines drove the SMARTBoard and horizontal DVIT displays and were responsible for running the main application.

The SMARTBoard is an upright, touch-sensitive, rear-projected display with a 167.6cm screen (diagonal). The DVIT display is similarly sized, but is front-projected. Both displays were set to display the workspace at a resolution of 640x480. Although the horizontal DVIT could technically support two simultaneous touches, the SMARTBoard could not. To prevent this technical difference from affecting the results of the study, the software was written to allow only one touch per board for the study.

For the DigitalArms conditions, participants were seated, and equipped with optical mice placed on the table. For the VideoArms conditions, participants were given yellow dishwashing gloves as their bright, uniform color provided better extracted arm images. While VideoArms was designed to pick up skin tones, configuring the system for each group would have been time-consuming, which was unacceptable since I had these participants for only a short time. Also, since my primary interest was not in the computer vision algorithm but rather in the collaborative aspects of the system, I felt this was a reasonable substitution.

The experimental setup for Stage 1, which used only a single person in each room is depicted in Figure 5.2. The setup for Stages 2 and 3 are depicted in Figure 5.3. In all cases, participants were allowed to move freely around the display.

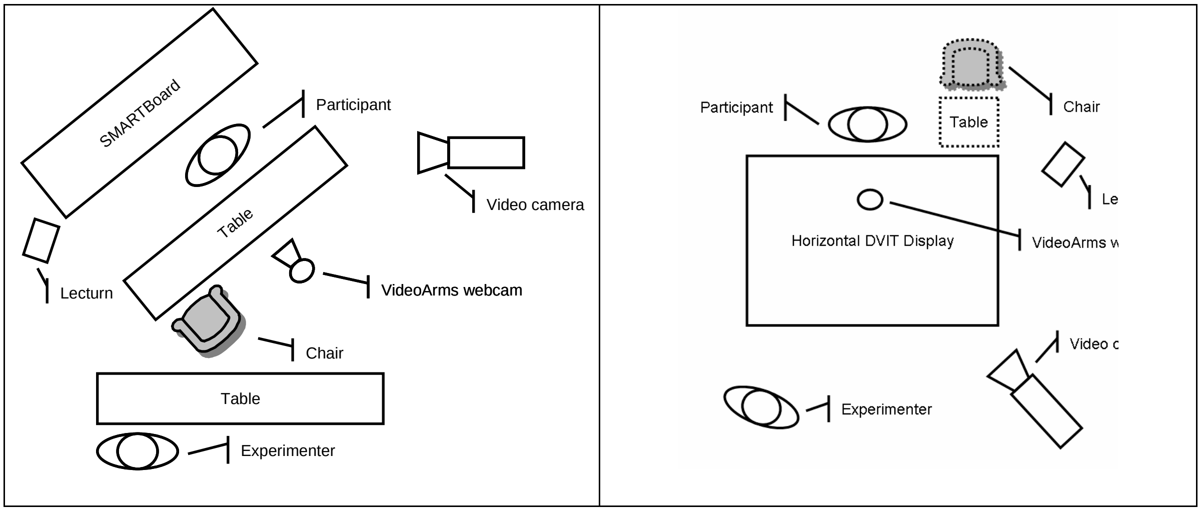


Figure 5.2. The experimental setup for stage 1 in: (a) the room with the SMARTBoard, and (b) the room with the horizontal DVIT. In the room with the DVIT, the table and chair were removed for the VideoArms conditions, but in place for the DigitalArms condition.

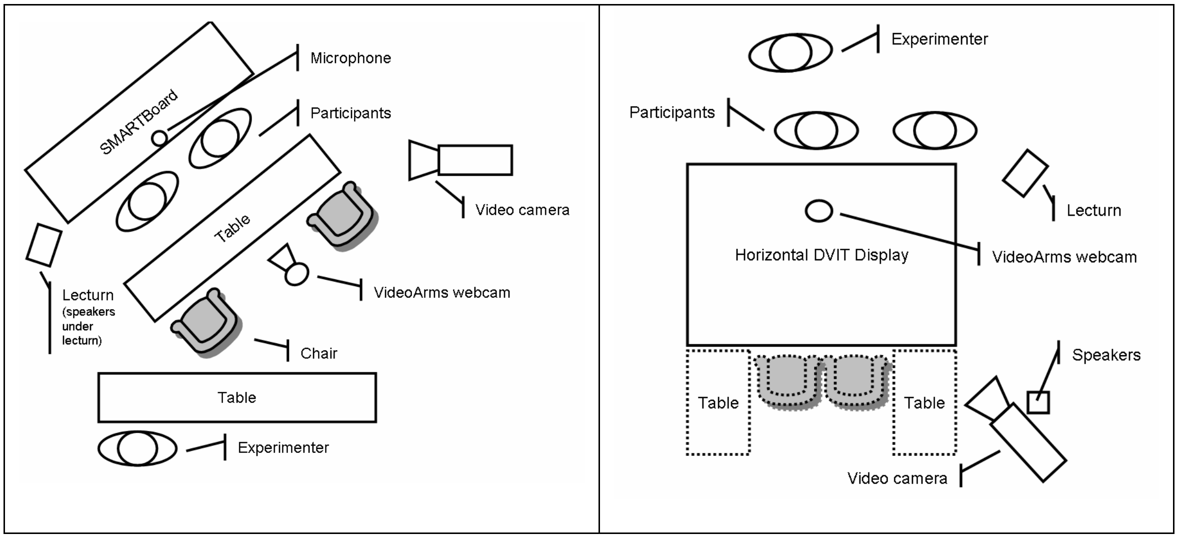
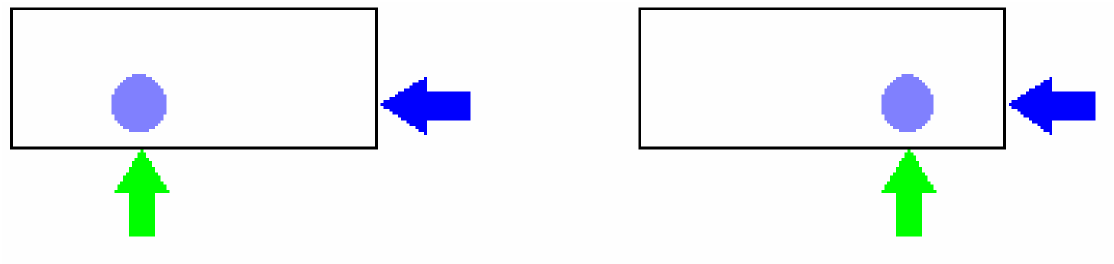


Figure 5.3. The experimental setup for stages 2 and 3 in: (a) the room with the SMARTBoard, and (b) the room with the horizontal DVIT.



.
Figure 5.4. The large dot is a cooperatively controlled object. Movements by the two cursors (controlled by different participants) change the dot's position. Adapted from Bricker, Baker & Tanimoto (1997).

Skype, a peer-to-peer PC telephony application, was used to provide a voice link between rooms in Stages 2 and 3 of the evaluation. This software produces telephone (or better) quality audio with a 0.5s delay. In both rooms, the microphone was placed above the display while the speakers were placed on the side

5.2.4 Tasks

Depending on the stage, participants performed one or both of two different tasks: the elastics and nails task and an open-ended design task.

5.2.4.1 Elastics and nails task

The elastics and nails task was designed specifically to evaluate the effectiveness of VideoArms in supporting gesturing and consequential communication. It was also designed to avoid a known difficulty in evaluating groupware: because groups are highly resilient, they can almost always adjust strategies to work around poorly designed systems. For instance, a common participant strategy is divide-and-conquer, where participants independently complete an agreed upon portion of the task with minimal interaction (e.g. Tse, Histon, Scott & Greenberg, 2004).

To circumvent this strategy, I wanted a task that required closer cooperation and interaction between participants. The solution was to use cooperatively controlled objects (Bricker, Baker & Tanimoto, 1997), a class of objects that demands simultaneous interaction by multiple participants (Figure 5.4). Figure 5.5 shows two examples of

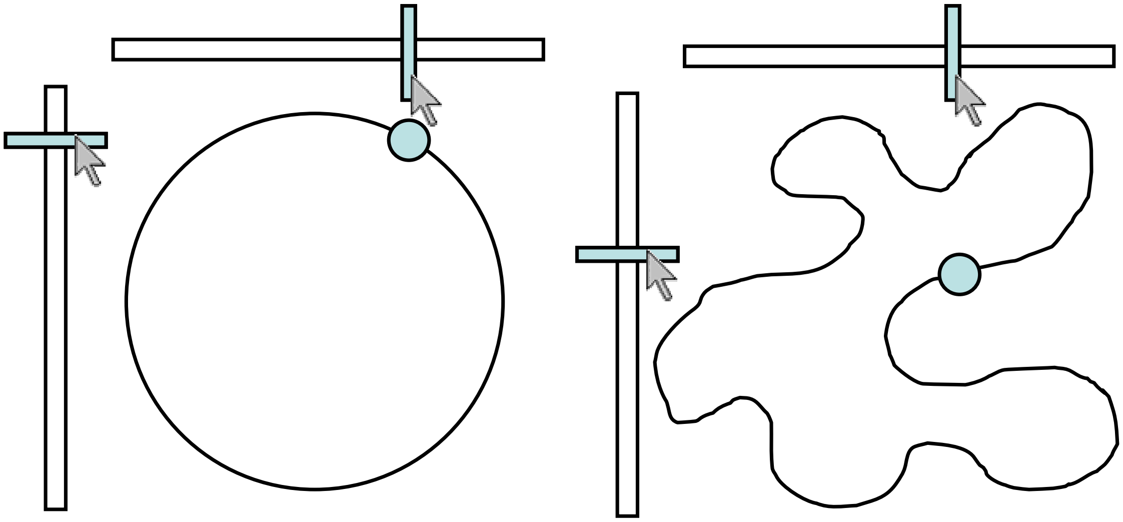


Figure 5.5. Two cooperatively controlled tasks. On the left, two players control the motion of a dot around a circle. On the right, two players control the motion of a dot around a more complicated figure. Adapted from Bricker et al., 1997.

cooperatively controlled tasks. In each, one user controls the X coordinate of a ball, while the other controls the Y coordinate of the same ball. Together, the players must navigate the ball (a cooperatively controlled object) along the specified path as quickly as possible. While these tasks may appear artificial, they mimic the real-world collaborative activities (e.g. picking up a heavy couch and navigating it to a nearby exit).

The elastics and nails task is a drawing activity with multiple cooperatively controlled objects. Participants construct images (e.g. Figure 5.6a) with several randomly placed lines (elastics). Each elastic is a rubber band with two grab points (nails) that highlight when grabbed (Figure 5.6b). To move an elastic, participants must grab opposite nails of the same elastic, whereupon it can be moved by simply dragging the nails around. If either party releases his or her nail, the elastic sets itself down in place. Simultaneous interaction with the elastic is required to move it. If only one party grabs the elastic, the nail highlights itself, but the elastic will not move.

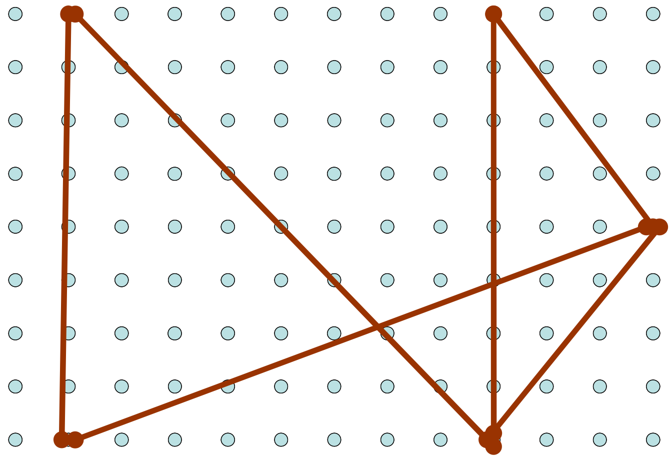


Figure 5.6a. An elastics and nails target image.

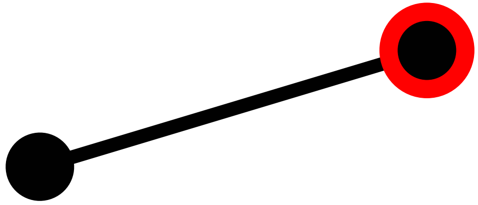


Figure 5.6b. An elastic with nails on each end. The right nail has been selected, and is therefore highlighted with a halo.

5.2.4.2 Open-ended design task

The second task was an open-ended drawing task with a whiteboard-like application where participants could simultaneously sketch and erase. Participants were asked to design and sketch the user interface of a print dialog for digital photographs.

I included the design task to determine whether the behaviours I would see in the elastics and nails task would occur in a less constrained scenario. Since the design task is, by nature, a free form activity without low level constraints, behaviours in this task could more generally be attributed to the embodiment technique itself rather than the task.

5.2.5 Procedure

Each stage of the evaluation uses variations of a common protocol. The specific protocol for each stage is described below.

5.2.5.1 Stage 1 procedure

This stage uses groups of 2 participants. Participants were introduced to the operation of the displays and the elastics and nails tasks. One person was then led into a room with a SMARTBoard display (Figure 5.2a) while the other was led to the horizontal DVIT display (Figure 5.2b). Participants then cooperatively controlled an elastic until they felt comfortable with both the display and the task. The two rooms were *not* connected with any sort of audio link; thus, the only method for communication was system feedthrough and the embodiment itself.

The focus of Stage 1 was the elastics and nails task using VideoArms and DigitalArms. Each trial had six elastics, chosen to optimise for time. Participants were asked to use a think-aloud protocol, a well-known protocol that offers observers insight into participants' thoughts as they progress through a task. To further encourage interaction, only one participant was given the target image (e.g. Figure 5.6a). This participant, called the *director*, ensured that the resulting image, cooperatively constructed with the *follower*, matched the target image. Since I was primarily interested at this stage in whether VideoArms could support gesturing and consequential communication, this *director-follower* paradigm was chosen to force a certain level of communication between the participants. The trial was deemed complete when the *follower* signalled the experimenter, thereby necessitating at least one explicit, interpretable signal from the *director*.

Participants completed eight trials: two as leader and two as follower with each embodiment technique. Participants then completed a questionnaire to collect preference and demographic information, and a semi-formal interview to elicit feedback about the system. Finally, they were debriefed and paid.

5.2.5.2 Stage 2 and 3 procedure

The procedure for Stage 2 and 3 were identical, and both used groups of 4 people. Similar to Stage 1, the groups of four were introduced to the touch screen displays and the task. They were then separated into groups of two: one pair would work in front of the

SMARTBoard (Figure 5.3a) while the other pair worked over the horizontal DVIT (Figure 5.3b). This time, however, the rooms were linked with the digital audio channel.

Participants completed two elastics and nails trials: one with DigitalArms and one with VideoArms. In this case, each trial had a total of 20 elastics since I was primarily interested in whether gestures would still be used given the audio channel. Prolonging the duration of each trial provided more opportunities to observe such occurrences.

As in Stage 1, only one participant on each side of the link was given a copy of the target image. Again, these participants were tasked with directing the action for the followers with the stipulation that directors were *not* to touch the elastics themselves. Participants followed with one embodiment technique and directed with the other.

Once the trials of elastics and nails were complete (one in each condition), participants moved onto the open-ended design task where they had to design and draw a print dialogue for digital photographs. Participants were given a copy of the instructions to read and refer to as they completed the task with VideoArms only. When they felt the task was complete, they signaled the experimenter. Participants then completed a questionnaire and a semi-structured interview before being debriefed and paid.

5.2.6 Design justification

The study as described above is not intended to be a controlled experiment facilitating statistical decision making. Rather, it is a fairly broad-brush observational study where I am primarily looking for occurrences of large effects. This approach is reasonable because at this early design stage, I am interested in validating whether VideoArms matches its theoretical potential as an effective embodiment, or understanding if there are specific problems that hinder it from achieving its promise. That said, the tasks need some justification, particularly in terms of their external validity (the extent to which the findings from the task can be generalized in the real world).

The elastics and nails is a contrived task; thus, its face-value external validity is suspect. That is, I would be extremely surprised to find a real world task with the constrained interaction of elastics and nails. However, the communicative acts and collaborative processes generated by this task are frequently found in real life workspace tasks.

- The transfer of information (in a director/follower way) is a common collaborative activity (Gutwin, 1997).
- Complex gesturing (beyond pointing) is a common workspace activity (Tang, 1991).
- The maintenance of awareness is a necessary requisite to collaborative activity (Gutwin, 1997).
- Consequential communication, as supported by the two embodiment conditions, aids in the maintenance of awareness.
- At times, simultaneous joint activity is required by two parties (Clark, 1996).

The use of the director/follower paradigm also appears suspect; however, consider that collaborators rarely go into meetings with equal sets of information. Often, one party has some information that is to be disseminated. In this case, the director plays that role, and disseminates that information to the followers using a combination of gestures and voice instructions. The director/follower relationship simultaneously gets at two of the three tasks from Gutwin's methodology (1997): following, and directing. In particular, followers will *follow* the director through the workspace (physically); similarly, it is likely that the directors will (verbally) *direct* followers through the workspace.

In the design task, the specific task of creating a print dialog for digital photographs is perhaps not common in the real world. Yet, it typifies generative design tasks, which are common in any brainstorming activity or task where ideas emerge or are refined pictorially over time. Examples of such tasks include the design of the information flow through an application, or the structuring of a diagram showing different parts of a business' supply chain, and all utilise the basic processes that this task requires: communication, collaboration, and content creation. This is why they have been used as the basic task in a variety of other studies (e.g. Minneman & Bly, 1991; Tang, 1991)

In generative design activities, collaborators must collectively generate ideas, suggest and draw design elements, and decide how different elements fit together. They do this primarily by communicating both verbally and non-verbally, through drawing marks that comprise image components, and by creating textual lists of ideas (Tang, 1991). As they do this drawing, collaborators monitor and coordinate each others' actions to ensure the final product accords to their expectations (Gutwin, 1997).

5.2.7 Data collection

Data was collected in several different ways. Since the evaluation was observational in nature, I sat in on all the sessions and took note of participants' behaviours and activities as they used the shared workspace. Next, I collected questionnaire data, conducted semi-structured interviews, and recorded task/trial time. All the data was analyzed together to gain a full picture of participants' experiences with the embodiment techniques.

5.3 Results

In this section, I report and discuss the study results by combining the results from all three stages of the evaluation to present a holistic view of the findings. I again emphasise that this study was not designed as a controlled experiment, and that the number of participants used was too modest to apply statistical hypothesis testing. While this makes my claims tentative, I stress that the behaviours I observed across the 22 participants in seven groups were fairly consistent, and thus suggestive of generalizable behavioural patterns.

Observations of the participants were coded using an open coding technique. Each observation was coded with a characteristic label based on the four principles from Chapter 3. If an observation fell outside these characteristic labels, I created a new label (e.g. "strategy")[2]. This coding methodology highlights behavioural similarities across groups,

[2] Although an open coding methodology was used, I only report on one new label here as it is the most pertinent to this investigation.

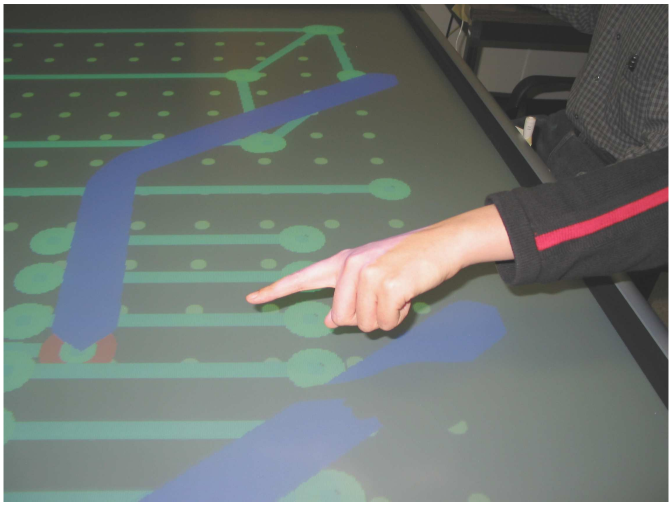


Figure 5.7. Participants often pointed with their physical hands as opposed to using the DigitalArms.

and identifies idiosyncratic behaviour of particular groups (Neustaedter, Elliot, Tang & Greenberg, 2004).

5.3.1 Gestures

Participants used gestures extensively with both prototypes, though the nature of these gestures differed. With DigitalArms, gestures were motion-based: the semantic content of the gesture was embedded in its motion; in contrast, I observed a mix of static and motion-based gestures with VideoArms.

With DigitalArms, gestures I observed in Stage 1 of the evaluation included: waving to indicate presence, shivering (slight, rapid movements back and forth) to attract attention, a directed thrusting to indicate a point, and a tickling motion (a playful touching of the remote arm) to indicate the completion of the task. While these gestures are based on real-life gestures, they also suffer from a level of indirection due to the constrained nature of the embodiment.

In the Stages 2 and 3 of the evaluation involving two groups of two, I also observed collaborators "incorrectly" pointing with their physical hands instead of using the DigitalArms (e.g. Figure 5.7). This meant gestures would not be seen by remote

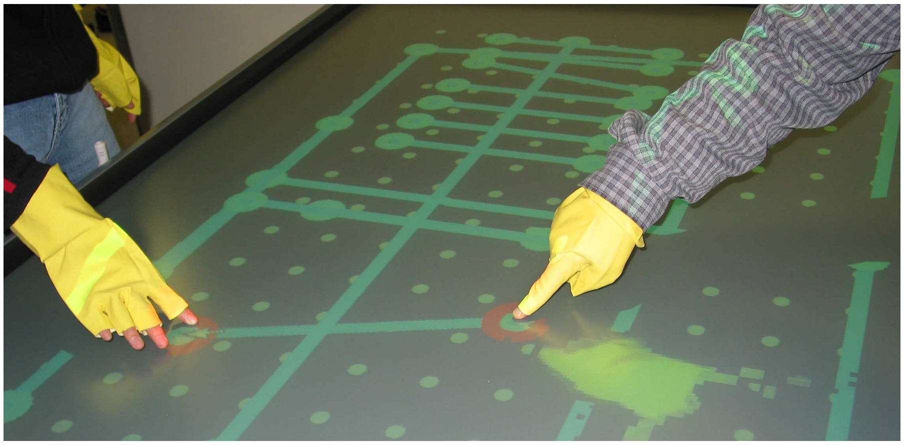


Figure 5.8. Participants commonly made use of pointing gestures.

participants. These instances suggest that participants generally preferred the naturalness of gesturing with corporeal hands and arms over DigitalArms for collocated partners.

In contrast, VideoArms supported much more natural, easily interpreted gestures (e.g. Figure 5.1 and Figure 5.8). Because Stage 1 participants did not have an audio channel, gestures often acted as audio substitutes: waving to say hello, waving as a way of saying, "push it that way", or "bring it this way", an a-okay sign, a hold gesture (open hand with fingers apart), exaggerated clapping as a "we're finished" signal, an open-handed wave as a signal to stop or to say something was wrong, and a thumbs-up to signal that something was correct (recall that in stage one, there was no voice channel). Over all three stages, the variety of VideoArms gestures observed was fairly extensive. From the Bekker et al. (1995) taxonomy, all three workspace-related gestures were observed (kinetic, spatial, pointing). Beyond these, we observed deixis (referential gestures relating to speech), as well as emblems. Comments with respect to gesturing with VideoArms were positive, e.g.:

> *C: "I liked VideoArms because it feels more natural. I can signal her the way I normally would with this sort of sign language."*

However, because the fidelity of VideoArms was low, participants generally exaggerated the nature of these gestures both in speed and in size. One possibility for why motion-based gestures were expressed slowly is that the frame rate for local feedback was fairly low (12 fps): two participants reported that they consciously slowed their actions so that

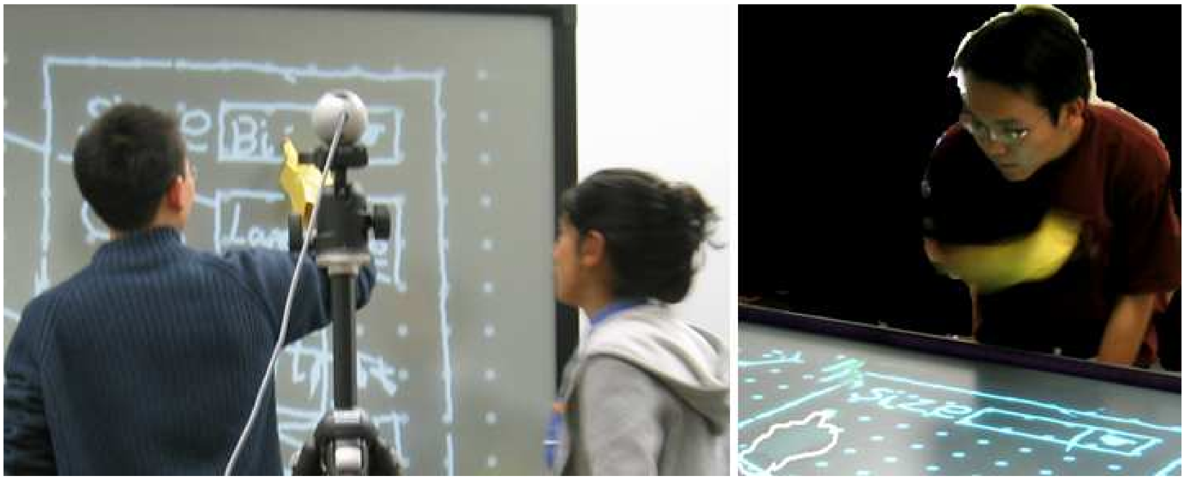


Figure 5.9. Participants spent a lot of time watching each other. On the left, H watches her collocated partner, W's activities. On the right, D also watches W's activities carefully via VideoArms (W's hand is outlined in white for clarity).

their gestures could be interpreted with the reduced frame rate. To test this theory, Stage 3 ran one group of four at a faster frame rate (20-25 fps), achieved by removing local feedback. In this condition, all four participants gestured more rapidly (in general) compared to participants in other stages of the evaluation. While this result is promising, I caution that it is preliminary since only one group was observed.

VideoArms provided a remarkably useful medium for participants. Participants were able to fluidly gesture and integrate those gestures into their interactions with collocated and remote participants. The gestures seen were considerably more varied and natural than those expressed with DigitalArms.

5.3.2 Consequential communication

Participants spent a considerable amount of time observing their partners (both collocated and remote) to understand the state of the activity, *regardless of the embodiment technique*. In elastics and nails tasks, directors in Stage 1 would watch to ensure their partners had grabbed the correct nail, as well as to ensure their partners had positioned the nail in the correct location. When directors detected an error (e.g. if the follower grabbed the wrong elastic or moved a nail to the wrong location), directors would redirect followers to the correct elastic or location. Followers would reciprocally watch directors' actions to determine which nail to pick up. Over the course of the task, directors generally would not even gesture at which nail to pick up as the consequential communication sufficed.

I also observed several instances of corrective activities (correcting another's actions in the workspace) that occurred across the link, which is facilitated by consequential communication. In the elastics and nails task, it may be to place an elastic in the correct location. In the open-ended design task, it may be to put a picture element in a different place. Corrections are predicated on understanding the state of the workspace and the activities of other participants—knowledge that is gained primarily through consequential communication. If an embodiment supports consequential communication, we should therefore expect to see corrections occurring across the distributed link. In both Stages 2 and 3 of the evaluation, we saw various instances of correction occurring across the link. For instance, one participant interfered with a remote participant by waving aggressively:

> *R: I was mostly watching F, but I could also see what M was doing. When it seemed like M wasn't doing what we'd agreed on, I asked him what the heck he was doing.*

When probed about what he had been watching, R responded:

> *R: At first, it wasn't so much what he was drawing. I could see that he was completely in the wrong place. When he started drawing, I knew he had the wrong idea.*

Yet the consequential communication provided by the embodiments was not perfect. For example, participants remarked that the imprecision in the video quality of VideoArms often made it difficult to understand their remote partners' activities at times. The problem of image noise (as discussed in Section 4.3.3) meant that the specific location of remote participants was not clear. Furthermore, because remote participants were represented as yellow latex gloves, it was difficult to identify who was doing what at the remote display. In contrast, DigitalArms provided a grounding point (the shoulder) as well as a very definitive point location. Thus, although VideoArms gave users the freedom to move around the workspace, this freedom actually caused problems in identifying remote collaborators arms (one user remarked, “It’s hard to figure out who is who when they’re moving around all the time”), and therefore problems in determining the activities of specific individuals.

In spite of VideoArms’ poor video quality, participants made constant use of the embodiment as a source of consequential communication. Both VideoArms and DigitalArms helped to increase engagement as evidenced by the incidents of corrective acts across the link.

5.3.3 Local feedback

Participants’ responses about the local feedback provided by VideoArms were mixed, reflecting the mixed usage of local feedback during the study. Some participants readily acknowledged the utility of the feedback while others questioned its value; however, an implementation issue may have been the cause of some participants’ rejection of local feedback. In contrast, participants did not mention the feedback provided by DigitalArms at all.

On the positive side, some participants made heavy use of local feedback to gauge the interpretability of their gestures. For instance, some participants working in front of the upright display would stand on either side of the display (as opposed to directly in front of it) to give the camera an unobstructed view of his/her arms (Figure 5.10).

> *C: “I really liked being able to see myself. At first, I wasn’t sure if J could see what I meant, then I realized I could see myself.”*

In contrast, other participants made no use of the local feedback, some because the camera placement limited the utility of feedback, and others because they failed to see the utility of it. Participants were essentially unable to make use of feedback on the front-projected display since the feedback was projected directly onto their arms. In one particularly memorable episode, H’s pointing gestures were blocked from the camera’s view by her partner W (Figure 5.11).

> *H: You need to move it here. Pick this one up. [Gesturing]*

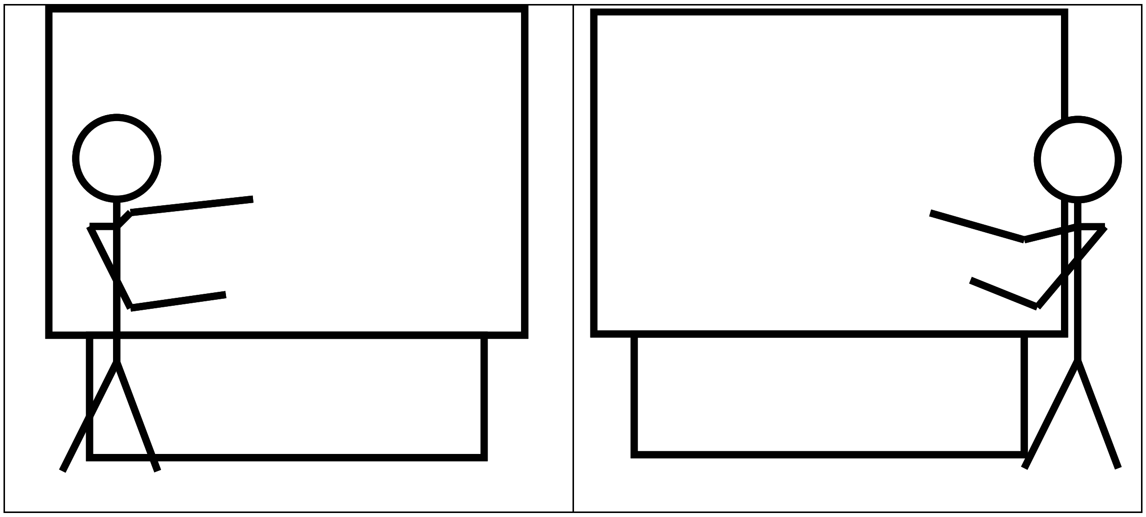


Figure 5.10. Participants would stand on either side of the whiteboard, contorting their bodies in unnatural ways so that their hands could be captured by the camera.

D (remote collaborator): This one?

H: No this one here. Look! [Gesturing]

D: Huh?

H: Look at my hands! [Waving]

D: Wait, you are moving too quickly. Slow down. I will tell you when I see you.

This confusion lasted for over 30s. It was only resolved by trial and error on D's part (sequentially picking up elastics until H verbally acknowledged his action). The group never figured out that the source of their frustration was the positional problem posed by VideoArms' camera capturing technique.

Other groups remarked about the apparent lack of utility of the local feedback.

A: "I can see myself just fine. It doesn't make sense to draw it on the screen. [It] seems like a waste."

Finally, as detailed in Section 5.3.1, participants generally gestured slowly with VideoArms because, based on the local feedback, they believed (correctly) that their actions would be sent to remote participants at a very low frame rate. As a result, participants slowed their gestures down so they could be interpretable. In the single trial of Stage 3, I removed the local feedback altogether without any ill effects: participants did not note the lack of feedback.

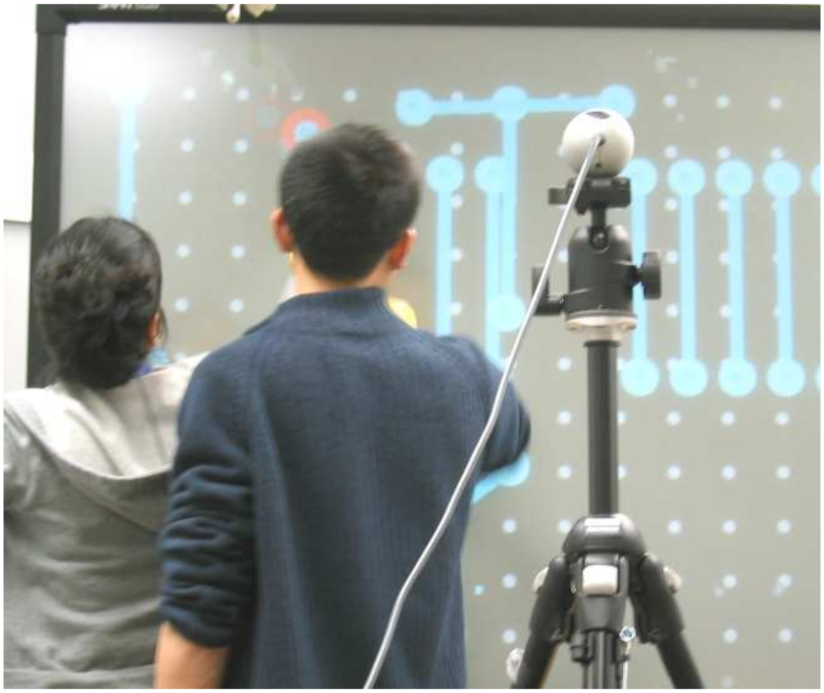


Figure 5.11. Some groups made no use of local feedback. In this case, H (left) is the director. H's pointing gestures (made with her right hand) cannot be seen remotely because the camera's view is blocked by her partner, W (right).

The utility of the local feedback provided by VideoArms was not effectively evaluated in this study due to camera placement requirements of VideoArms (as discussed further in Section 5.4). At this point, it seems clear that some participants benefited and clearly made use of local feedback while others did not use local feedback at all. Finally, the single trial of stage three suggests that local feedback may play an implicit role in participants' workspace behaviour.

5.3.4 Elastics and nails strategy

Groups generally used piece-by-piece strategy to complete the elastics and nails task, completing the movement of one elastic before moving the next elastic. For pairs in the first stage of the evaluation, I observed two sub-strategies: *passive follower* and *active follower* strategies. For the latter two stages of the evaluation, groups adopted either a *tiered director* or *peer director* strategy.

In the *passive follower* strategy, the director essentially completes the entire task alone. The director picks up a nail, and the follower picks up the corresponding nail of the elastic, allowing the director to move his or her nail to the correct locations. The participants then switch nails, allowing the director to move the second nail into place.

In the *active follower* strategy, the director gives directions to the follower, and relies on the follower to move one nail appropriately while moving his or her own nail. The execution of this strategy differed with VideoArms and DigitalArms. With VideoArms, the director would use one hand to grab a node, and use the second hand to point at the destination location for the learner. With DigitalArms, groups adopting this strategy would generally highlight the follower's nail (by alternately selecting and deselecting the nail, thereby making the halo flicker on and off), and then gesture at a destination location for the nail. Then, both the follower and director would grab their respective nails, and move the elastic accordingly.

Groups of four used two general strategies: a *tiered director* and *peer director* approach. The *tiered director* approach involved designating one director as the primary director. The primary director was responsible for identifying which elastic to select, as well as directing *both* followers to their target locations. The subordinate director was responsible primarily for ensuring that the primary director's orders were carried out. Specifically, the subordinate director was responsible for ensuring that his or her collocated partner's nail was in the correct location.

The *peer director* approach was slightly more democratic in that both directors worked closely together. Together, the directors would select an orientation for the image, divide the image into its semantic components (e.g. "piece that looks like a rocket ship", "the special area", "that house-like thing"), and select an order in which the image would be constructed.

The latter approach was generally more chaotic than the tiered approach since it required on-the-fly negotiation. Conflicts often arose with respect to which elastic in the image was being constructed, as well as the orientation of the image. These conflicts needed to be resolved before the group could proceed. As a result, many teams who began using a peer director approach switched to a tiered director approach halfway through the task.

5.3.5 Task time and preference

I collected data on some comparative usability measures to get a general sense of the usability of VideoArms. I caution against gross generalisations based on this data since the number of groups were low, preventing statistical decision making. In general, preference data suggests that VideoArms is perceived more poorly than DigitalArms primarily because of its implementation; however, objective measures suggest that it is certainly no worse than DigitalArms.

In the first stage of the evaluation (pairs), a total of 24 elastics and nails trials were completed by the three groups. Task time for trials using VideoArms ranged from 83 seconds to 529 seconds (average task time was 219 seconds, with a standard deviation of 121 seconds.) compared to DigitalArms, which ranged from 70 seconds to 212 seconds (average task time was 132 seconds, with a standard deviation of 41 seconds). In this sample, DigitalArms outperformed VideoArms in task time, though more participants preferred VideoArms (4 vs. 2).

An implicit of measure of usability, the Subject Duration Assessment (SDA), gives perceived task completion time as a percentage difference from the actual task completion time (Czerwinski et al., 2001). Positive values on this measure indicate that users find the task easy to complete (ostensibly due to the user interface); negative values on this measure indicate that users find the task more difficult. For DigitalArms, the average perceived trial completion time was 138s, an SDA value of -4.6%. For VideoArms, the average perceived trial completion time was 173s, and an average SDA value of 26.4%. Thus, VideoArms was the more usable interface technique on this implicit measure of usability.

Groups in Stage 2 of the evaluation (groups of 4) also completed the elastics and nails task more rapidly with the DigitalArms (summarised in Table 4.1). On average, completing the task with DigitalArms took 556s compared to 754s for VideoArms.

Group	DigitalArms task time	VideoArms task time
1	874	870
2	365	484
3	430	907

Table 5.1. Stage 2 task times for elastics and nails task (in seconds).

Participants reported preferring DigitalArms over VideoArms in the questionnaire (13 vs. 3); however, in the post-study interview, over half the participants reported that they would have chosen VideoArms had it been implemented better. That is, while VideoArms was a good concept, its implementation proved limiting. Comments include:

> *"It was too blocky, so it's hard to tell what's going on."*
>
> *"There are too many extra dots. I can't see properly."*
>
> *"Mice are more convenient. You have to move too much."*
>
> *"It should be easier for the layman [to use VideoArms]. It's much faster with hands."*
>
> *"I like it, but it is too slow."*
>
> *"It's more natural when I talk to [remote participants] to use the gloves."*

The SDA for Stage 2 participants rated both interfaces as less than optimum: DigitalArms scored -12.4% while VideoArms scored -8.62%. This means that participants generally overestimated the time they spent using each embodiment technique to complete the task.

In contrast, Stage 2 participants generally overestimated the task time for the design task (actual times in Table 5.2), with an average SDA of 15.6%. This score tells us little beyond the fact that participants enjoyed the design task using VideoArms since we cannot compare it against a similar, DigitalArms version of the task. However, one interpretation is that when used in a more open-ended task, VideoArms was liked because it provided participants with the opportunity to more freely express themselves.

The group of four in Stage 3 of the evaluation (higher frame rate for remote participants) completed the elastics and nails task with VideoArms extremely quickly: 458s. Similarly, the design task (with VideoArms) was completed in 385s. One

Group	Design task time
1	1200
2	540
3	915

Table 5.2. Stage 2 task times for the design task (in seconds).

interpretation of these values is that the higher frame rate meant that participants could gesture more rapidly and therefore complete the tasks quicker. However, this group was in general, much quicker than other groups, completing the elastics and nails task with DigitalArms in 217s. All four participants in this group preferred DigitalArms. SDA values for this group were similar in flavour to the groups in stage 2 (DigitalArms: -59.7%, VideoArms: -1.57%, design task with VideoArms: 7.0%).

Participants' reactions to VideoArms at a preference level were mixed: they preferred DigitalArms over the buggy implementation of VideoArms. In this sample, task time generally supported DigitalArms; however, the implicit measure of usability, subject duration assessment, suggests that VideoArms was the better interface technique. In sum, VideoArms certainly did not fare worse than DigitalArms; however, it is hard to make conclusive decisions based on this small sample.

5.3.6 Addressing presence disparity

VideoArms was designed to address presence disparity in mixed presence groupware systems. This evaluation was designed to determine whether VideoArms mitigated presence disparity. To answer that question, I bring the discussion back to the set of questions I introduced in Section 5.2.1. Each stage of the evaluation had a set of questions that I was interested in, and I address those here.

In Stage 1, participants completed the task in pairs and had no voice link. In this case, it was very clear that participants made use of VideoArms to gesture within the context of the workspace. I observed deixis (gestures accompanied by speech and referring to a location or an item in the workspace, e.g. “This one”), which generally have no interpretation out of context. I also observed a wide variety of natural gestures with VideoArms, although they were exaggerated both in time (they were executed slower) and in space (they were often grandiose gestures). Participants also made use of VideoArms by carefully watching the arms of others in the workspace, since in Stage 1, this was the only means of communication. Finally, participants made use of local feedback, in many instances contorting their bodies so that the camera would see their arms (Figure 5.10).

In Stage 2, I increased the group size to four, and added a voice link. In this case, I still observed gestures in the workspace, many of which were intended for collocated participants, but sometimes also intended for remote participants. Importantly, gestures *were not replicated* for remote participants: a single gesture was generally sufficient to communicate to both collocated and remote participants. I also observed consequential communication across the link, and saw that it facilitated error-correction.

Finally, in Stage 3, I removed local feedback to increase the frame rate of the embodiments. I saw that gestures were conveyed at a much more natural pace, though they were still exaggerated in size. In my sample, the group in this last stage of the evaluation completed tasks much more rapidly than groups in Stage 2, lending preliminary support to the idea that rapid gesturing increases the speed of collaboration.

It seems that participants were able to discover and make use of the features provided by VideoArms. Yet, did VideoArms, by providing some of the features of a physical embodiment, reduce presence disparity? Recall that I defined presence disparity as a marked difference in the level of engagement between collocated and remote participants. I suggested that by increasing the fidelity and richness of the embodiment of remote participants, we could increase the level of engagement with remote participants. In this evaluation, we observed that the increased fidelity of VideoArms enabled richer channels of communication, both by explicit and varied gestures, and with consequential communication. In Stage 1, for instance, we observed DigitalArms’ gestures to be more ambiguous and less varied than with VideoArms. Similarly, I observed many instances of across-the-link engagement with respect to error-correction—a direct result of consequential communication. Thus, it is clear that VideoArms indeed reduced presence disparity.

Yet, I hardly consider the results of this evaluation spectacular. Two key measures, one of task performance, and one of preference, indicate that VideoArms in its current state, still leaves much to be desired.

5.4 Discussion

The results of this study confirm primarily three statements. First, people use complex gestures in the workspace, and like doing so. Second, the current VideoArms implementation is an imperfect system that requires retooling. Finally, although VideoArms was a crude implementation, it increased both communication between remote participants and the overall level of engagement across the link, thereby reducing presence disparity.

As Tang (1991) observed, a large portion of the workspace activity involving hands are intentional gestures intended to attract attention or to convey some idea. These gestures are natural and fluid, occur in everyday conversation, and have accepted meanings (e.g. the gestures in have common and accepted meanings in Canada). Beyond these simple static gestures, many gestures are also motion-based. For instance, moving one’s hand in one direction and repeating that action may indicate a “this way” gesture even though the posture of the hand and fingers have little bearing on the interpretation of the gesture. Given questionnaire and interview comments, VideoArms appears to support natural gestures in a manner much like what users desired and expected. Aside from being complex, gestures in the study were often combined in novel ways that would have been impossible to predict (for instance, I observed clapping behaviour to signal “good job”). And again, many participants made extensive use of the two-handed interaction provided by VideoArms.

In spite of this success, VideoArms had a key technical failure.

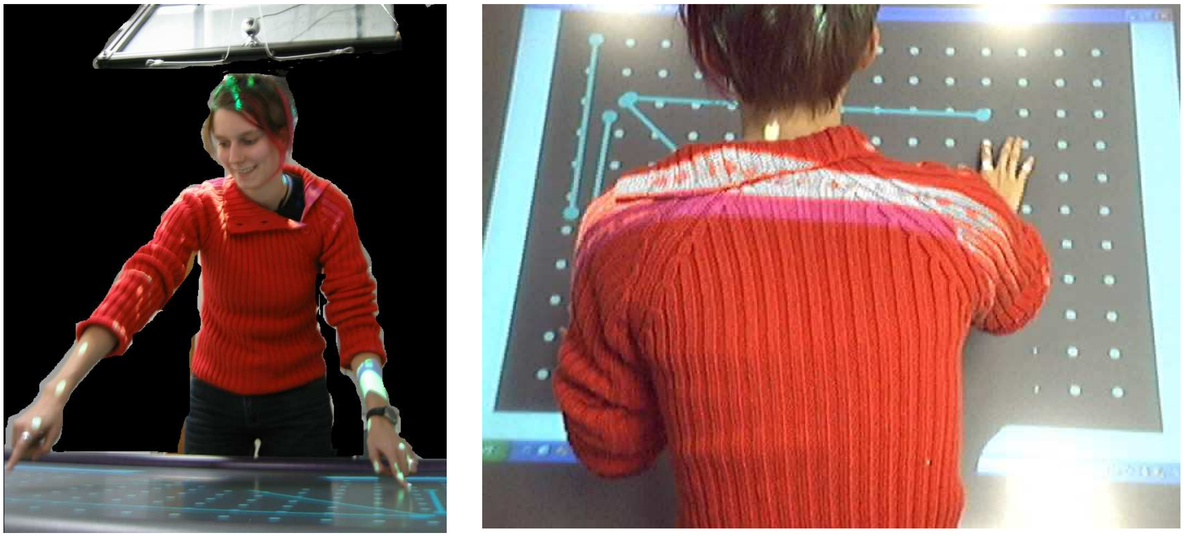


Figure 5.12. In this recreated photo, the VideoArms' camera angle (right) cannot capture the image of the participant's hands as she works in the work surface (left).

> *J: If the image quality were better, [the system] would be better. Sometimes, it wouldn't even show me properly.*

VideoArms' images were not clear and crisp enough for participants. First, the colour segmentation technique I used was not perfect, producing on-screen artifacts or holes and sometimes confusing users. Second, a participant's body working in front of the display could obscure the camera's view of the participant's arms (Figure 5.12). Finally, the camera's 320×240 image was blown up to 640×480, which meant that the image was not very crisp.

VideoArms clearly provided a very rich medium for participants' interactions with remote parties. Participants enjoyed using the medium to work with remote parties (as indicated by the SDA scores and questionnaire responses). It also provided a means for users to detect and correct errors by collaborators, indicating that it provided a rich medium for consequential communication. Finally, in comparison to DigitalArms, VideoArms provides a richer non-verbal communications medium, thereby increasing the overall level of interaction and engagement in the workspace.

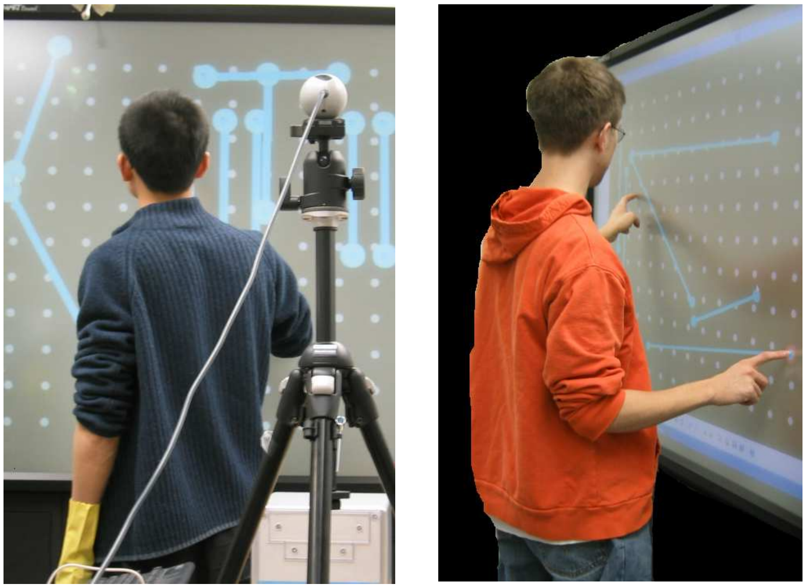


Figure 5.13. People generally like to work with their hands comfortably in front of themselves as opposed to outstretched.

5.4.1 VideoArms: For laboratory use only

The design principles behind VideoArms were formulated to target presence disparity, and the study presented here demonstrates that VideoArms, by supporting these four principles, mitigates presence disparity. In principle then, VideoArms is an effective embodiment technique for mixed presence groupware.

In spite of the successes of VideoArms, it is not a suitable technique in its current form for remote embodiment beyond the laboratory setting. VideoArms' main failing, beyond its actual software implementation (i.e. the computer vision algorithm and network transmission implementations), is the physical setup of the cameras with respect to the touch-sensitive board (Figure 4.3, Figure 5.2 and Figure 5.3). In my implementation, the cameras are placed approximately 2m from the surface. This distance allows the physical work surface to fill the entire camera frame, and the camera to capture the arms of collaborators as they work on the display. As mentioned earlier, however, this set up neglects the fact that people rarely work with their arms completely and uncomfortably outstretched in the workspace (as in Figure 2.6).

Instead, while people are interested in completing whatever task is at hand, people will endeavor to maintain comfort. For instance, when working with an upright display, people are more comfortable with their arms positioned in front of them (Figure 5.13) as opposed to contorting their bodies as in Figure 5.10. Similarly, when working over tabletop displays, people will rarely work by extending their arms far from their bodies; instead, they will lean forward with their bodies (Figure 5.12). The unfortunate consequence of these comfort-preserving behaviours is that the cameras no longer have an unobstructed view of participants' arms.

Thus, VideoArms is imperfect as a practical embodiment system because, while it maintains the naturalness of gestures and hand/arm-based communication in the work space, it forces collaborators to position themselves awkwardly.

5.4.2 VideoArms Constraints

Beyond the implementation itself, VideoArms is conceptually best suited for the constrained tasks that were used in the study. Specifically, VideoArms was designed for extremely literal environments whose artefacts are comparable in size to a fist, and 2-dimensional virtual work spaces. While the current implementation of VideoArms gives only fairly coarse input, techniques developed for pixel level accuracy could be used instead. Thus development of VideoArms for fine-grained interaction is certainly possible. Workspaces that represent abstract or three-dimensional information will benefit only modestly from VideoArms. Instead, more conventional approaches to embodiment may likely be considerably more effective in these scenarios. For example, Dyck & Gutwin (2002) suggest that avatars are more appropriate for three-dimensional workspaces.

5.5 Conclusions

I ran an observational study to determine whether VideoArms met all the principles I derived in Chapter 3 (second part of Goal 3). I demonstrated that VideoArms was a suitable embodiment technique for mixed presence groupware. VideoArms' main strength was mitigating presence disparity by eliciting natural, two-handed, complex gestures used for communicative purposes over the network link, and facilitating consequential communication, thereby increasing remote engagement.

While VideoArms certainly has its strengths, I also found many weaknesses in the system. The practical reality of how participants must position themselves to use VideoArms reduces the utility of the system for applications beyond the laboratory setting.

In the next chapter, I bring the discussion in full circle by reviewing the thesis problems that I set out in Chapter 1. I show how I addressed each of these problems, and discuss the future work that is implicated by this work. Finally, I conclude by discussing implications of this research within the larger context of groupware research.

Chapter 6. Conclusion

In this chapter, I begin by reiterating my four thesis problems from Chapter 1. I then describe how my work in this thesis contributes to knowledge by the way it addressed those problems. Finally, I discuss future directions this work could take by placing my contributions within the larger context of groupware research.

6.1 Thesis problems

In Chapter 1, I outlined three main problems I aimed to address in this thesis.

1. **We do not understand the technical and social challenges inherent in mixed presence groupware systems.** Because no documented MPG systems exist, I need to architect MPG systems myself. Although I can draw on existing literature to understand the dynamics of collaboration, I need to observe how the unique physical arrangement of users in an MPG system affects collaborative dynamics.
2. **We do not fully understand the role of embodiments in the specific context of mixed presence groupware.** Given my observations of MPG systems in use and existing literature on collaboration in shared visual workspaces, I need to understand how people use the visible aspects of their collaborators to facilitate collaboration.
3. **We do not know what embodiment techniques are appropriate for mixed presence groupware.** Because of Problem 2, we cannot yet build effect effective embodiments for mixed presence groupware. Indeed, what makes up an “effective” embodiment in mixed presence groupware is unknown.

6.2 Contributions

I addressed these three thesis problems in this thesis with the following research contributions.

1. **I provided a definition for mixed presence groupware, and an architectural pattern for its construction.** I defined a new area of research called mixed presence groupware, which addresses real-time collaboration by both collocated and remote participants. In Chapter 2, I described the implementation of MPGSketch, a shared sketching application supporting both collocated and remote collaborators. I decomposed the construction into an abstract architectural pattern for mixed presence groupware. This pattern will aid future researchers by guiding the architectural design of MPG systems, allowing researchers to focus their efforts on building functionality as opposed to being mired in low level details (Problem 1).

2. **I identified two problems unique to mixed presence groupware: display and presence disparity.** When upright displays are connected with horizontal displays, participants view the workspace from different orientations, a problem called display disparity. I also identified presence disparity, a problem where collaborators focus their collaborative efforts on collocated collaborators at the expense of their remote counterparts, and an imbalance that negatively affects collaboration (Problem 1).

3. **I provided a set of theoretical design requirements for embodiments in mixed presence groupware systems.** Because I was interested in mitigating presence disparity, I investigated the role of embodiments in the workspace, and their role in engaging remote participants. Based on the observable aspects of a person's corporeal body in a physical workspace, I developed a theory of embodiments for mixed presence groupware in Chapter 3. This theory highlights the importance of providing local feedback, support for timely gestures, support for consequential communication, and the placement of embodiments in the context of the workspace (Problem 2).

4. **I developed and evaluated VideoArms as a mechanism to mitigate presence disparity.** In Chapter 4, I focused on the iterative design and development of an

embodiment technique for mixed presence groupware called VideoArms. VideoArms is a video-based embodiment technique that captures and projects the arms of collaborators as they work on remote workspaces. The evaluation demonstrated weaknesses in VideoArms' implementation, but supported its design ideas of supporting gestures and consequential communication in remote workspaces.

6.3 Building on VideoArms

VideoArms was a fairly successful design idea given its prototypical nature, and the constrained domain of tasks for which it was considered. One of its strengths, which is implied by the non-tethered input system, is that the embodiment is created to mimic real-world collaborative phenomenology (Dourish, 2001). The local feedback provided by VideoArms enforces this false belief, giving the user a greater sense of control over the embodiment.

Parallel, but independent work in linked CAVE environments (e.g. Gross et al., 2003) has focused on this conception of embodiment as telepresence. Generally, the focus of these environments is to recreate the phenomenology of a spatial area. Blue-c creates an extremely rich three dimensional model rendering of collaborators in these environments to enhance telepresence of remote collaborators (Gross et al., 2003). The unfortunate drawback of this approach is the prohibitively high cost of the setups. Futhermore, it is unclear if 3-D environments are appropriate for doing the routine collaborative activities that one would expect to do on (say) a table top display e.g., sketching, drawing, collaborative document viewing and editing.

The VideoArms embodiments system itself is suggestive of two additional research steps. First, a more robust implementation is required. Secondly, a more formal study is required to validate its design. The implementation needs a more robust mechanism to capture the bodies of collaborators in the space. Using an infrared approach (Miwa & Ishibiki, 2004) to create the segmentation image (Figure 4.5, middle) combined with a regular camera would be an appropriate first step in a more robust implementation.

Successful extension of the VideoArms idea also requires a better and more detailed understanding of its successes and failures. A more formal evaluation would be required for this understanding, although this would only be done on a next generation version that fixes the obvious problems. This evaluation could be done in four steps. The first step would be to observe and gather information about the collaboration in a fully-collocated team. We would be gathering information such as how many interactions occur, how many words are spoken, to whom are these words spoken, and similar metrics for gestures. This step establishes some baseline metrics for how communication occurs in collocated groups. Once this baseline is established, we can then run several trials of similar tasks in an MPG environment without embodiments to establish numerical values for the presence disparity that occurs in this context (i.e. the task as well as the physical distribution of collaborators). In the third step, we would again run participants through the task, except with telepointer embodiments. In the final step, we then run a series of groups with VideoArms, and compare the quantified measures against both the fully-collocated, the no-embodiment, and the telepointer conditions. In doing so, we would be able to establish three things: the extent of "presence disparity", and the extent to which telepointers and VideoArms mitigate presence disparity.

6.4 Future work

The overarching goal of this research is to support distance separated groups of collaborators. By introducing mixed presence groupware, I have identified a key research area that has been addressed in part by distributed groupware researchers, and in part by collocated groupware researchers. Yet the challenges posed by mixed presence groupware are unique, and presence disparity in particular needs to be addressed within the appropriate context.

One of the problems distributed groupware researchers grapple with is that remote collaborators are not physically present. Ishii (1993) and Tang & Minneman (1991b) previously presented visionary ideas of how distance separated collaborators might be portrayed in remote workspaces (Figure 6.1 and Figure 5.6). Because the human body is a wonderfully rich vehicle for communication, researchers have, time and again, returned to the idea of presenting the body in remote workspaces. Yet part of the problem is that the body is so rich that we do not yet know what is important to transmit. Video-based systems try to send everything, while computer-based groupware tends to send only minimal information.

Most computer-based groupware, as I mentioned in Chapter 3, present location information via telepointers. Gutwin (1997) presents workspace awareness information with radar views, which can help provide information about other collaborators' activities. Some video-based groupware (e.g. Tang & Minneman, 1991b, and several others) focuses on transmitting the collaborators’ arms. Of course, we can do more than arms. Other researchers have focused on presenting gaze information about collaborators. Sellen, Buxton & Arnott (1992), for instance, developed a video-based multi-party voice-conferencing system allowing participants to establish and maintain eye-contact by spatial separating video devices (Figure 6.2, top). A quite different implementation achieves the same effect, where Vertegaal, Weevers & Sohn (2002) situates the video of collaborators in a virtual workspace to maintain gaze direction of people to objects within the workspace and between each other (Figure 6.2, bottom).

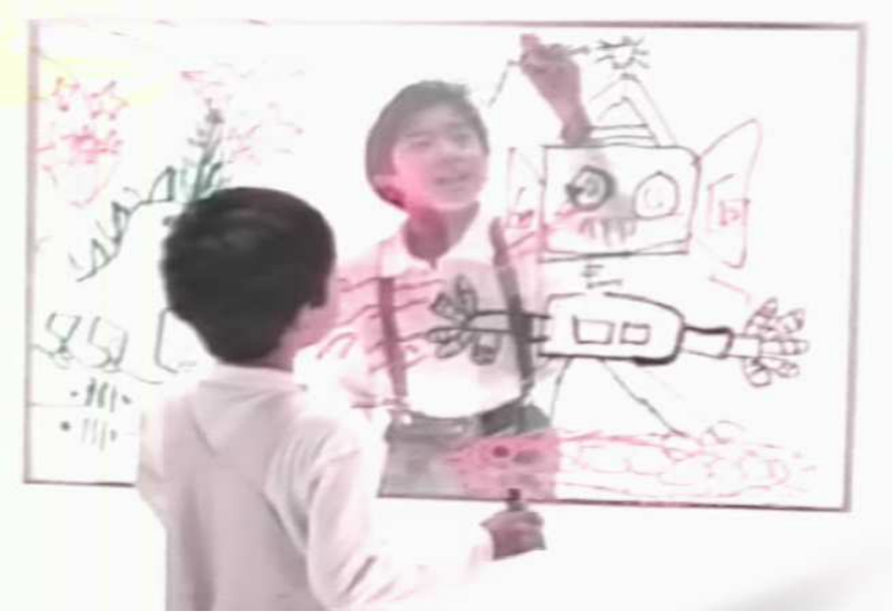
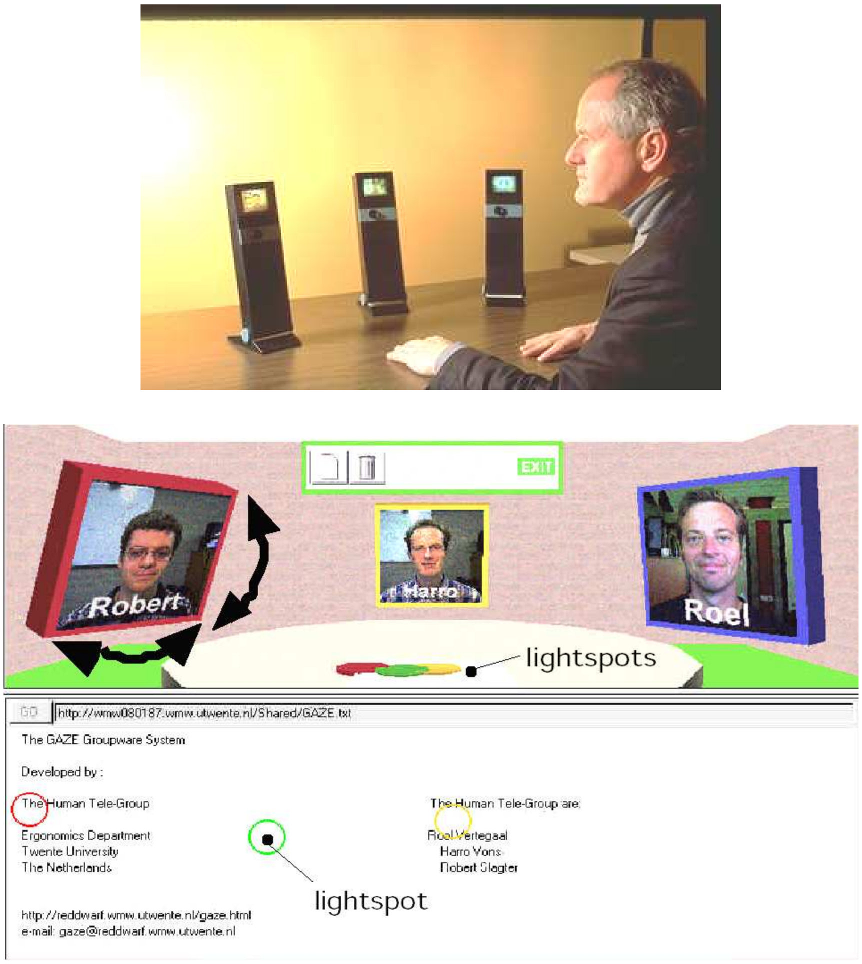


Figure 6.1. A vision of distributed collaboration. In this mock-up, the remote collaborator appears to be on the other side of a transparent window (Ishii, 1993).

Each of these approaches has demonstrated strengths; however, researchers have yet to orthogonalize these approaches to a single scale for designers, and to identify the aspects of the body that aid the different kinds of collaboration. For instance, which is more important, location information (i.e. what part of a workspace a collaborator can affect) or gaze information (i.e. what part of the workspace is a collaborator looking at)? The answer, of course, depends on situation and the collaborative task. In a transcription task, for instance, a collaborator may be looking at a source document, but is affecting the destination document. Another collaborator could potentially be interested in either of these documents. Thus, in the context of this research, a problem still to be addressed is the issue of how to virtualize and present a remote collaborator.



http://wmw080187.wmw.utwente.nl/Shared/GAZE.txt

The GAZE Groupware System

Developed by :

The Human Tele-Group

Ergonomics Department
Twente University
The Netherlands

http://reddwarf.wmw.utwente.nl/gaze.html
e-mail: gaze@reddwarf.wmw.utwente.nl

The Human Tele-Group are:

Roel Vertegaal
Harro Vons
Robert Slagter

Figure 6.2. Some researchers have focused on the importance of eye gaze in distributed collaboration. The Hydra system (top) presents each collaborator's image on each unit with a separate camera, thereby facilitating eye gaze (Adapted from Sellen et al., 1992). The GAZE system (bottom) shows how collaborators' virtualizations indicate gaze direction in the workspace (Vertegaal, 1999).

Collocated groupware research is still in its relative infancy, but already, problems relating to shared physical workspaces have begun to receive attention, such as territorality (Scott, Carpendale, & Inkpen, 2004) and orientation (Kruger, Carpendale, Scott & Greenberg, in press). Little work has investigated how collocated participants should be represented in the virtual workspace, which should not be surprising—the physical presence of collocated collaborators is representation enough. Yet collocated groupware supports collaborative dynamics beyond those afforded by distributed groupware. Further research in mixed presence groupware must ensure that, in supporting remote collaborators, the affordances provided by collocation are not compromised.

In the context of the present work, *presence* remains the key problem: remote collaborators are not present, and collocated collaborators are clearly present. The VideoArms embodiment solution presented in this work is one approach to the presence disparity problem: by presenting remote collaborators as they might appear in the workspace, we were able to better engage remote participants. Potentially, better image capture techniques (those that will not be affected by bodily positions of collaborators) and more efficient image processing would enhance VideoArms as I describe above; however, I believe presence disparity will continue to persist beyond what can be mitigated with technology alone. Socio-technical conditions, such as common ground, coupling of work, and collaboration readiness, will continue to effect presence disparity (Olson & Olson, 2000). These problems suggest that we should also consider addressing presence disparity by recognizing the social distance between remote groups (Bradner & Mark, 2002). This approach proposes that social, team-building tasks are also important for facilitating effective distributed collaboration.

6.5 Conclusion

In this thesis, I defined and developed mixed presence groupware systems, as well as an embodiment technique for mixed presence groupware. In doing so, I have identified a rich research area within computer supported cooperative work. Further work in this area will benefit distance separated groups of collaborators and users of groupware in general.

References

Apperley, M., McLeod, L., Masoodian, M., Paine, L., Philips, M., Rogers, B., and Thomson, K. (2003). Use of video shadow for small group interaction: Awareness on a large interactive display surface. In *Proceedings of the 4th Australasian User Interface Conference (AUIC '03),* (pp: 81-90).

Baecker R. (1992). *Readings in groupware and computer supported cooperative work.* San Mateo, CA: Morgan-Kaufmann.

Baecker, R., Grudin, J., Buxton, B. and Greenberg, S. (eds.) (1995). *Readings in human computer interaction: Towards the year 2000* (2nd edition). San Mateo, CA: Morgan-Kaufman.

Baker, K., Greenberg, S. and Gutwin, C. (2001) Heuristic evaluation of groupware based on the mechanics of collaboration. In M.R. Little and L. Nigay (Eds), Engineering for Human-Computer Interaction (8th IFIP International Conference, EHCI 2001, Toronto, Canada, May), Lecture Notes in Computer Science Vol 2254, p123-139, Springer-Verlag.

Bekker, M. M., Olson, J. S., and Olson, G. M. (1995). Analysis of gestures in face-to-face design teams provides guidance for how to use groupware in design. In *Proceedings of the ACM Coference on Designing Interactive Systems (DIS '95)*, (pp: 157-166).

Benford, S., Greenhalgh, C., Bowers, J., Snowdon, D., and Fahlén, L. (1995). User embodiment in collaborative virtual environments. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI '95)*, (pp: 242-249).

Bier, E. and Freeman, S. (1991). MMM: A user interface architecture for shared editors on a single screen. *Proceedings of UIST '91* (pp: 79-86). New York, NY: ACM.

Bly, S. A. and Minneman, S. L. (1990). Commune: A shared drawing surface. In *Proceedings of the Conference on Office Information Systems (OIS '90)*, (pp: 184-192).

Boyle, M. (2001). The effects of caputre conditions on the CAMSHIFT face tracker. Report 2001-691-14, Department of Computer Science, University of Calgary, Alberta, Canada.

Boyle, M. (2003). Collabrary shared dictionary v1.0.17: Programming paradigm and wire protocol. Report 2003-731-34, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.

Boyle, M., and Greenberg, S. (2002). GroupLab Collabrary: A toolkit for multimedia groupware. In J. Patterson (Ed.). *ACM CSCW 2002 Workshop on Network Services for Groupware*, November.

Bradner, E. and Mark, G. (2002). Why distance matters: effects on cooperation, persuasion and deception. In *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, (pp: 226-235).

Bricker, L.J., Baker, M.J., and Tanimoto, S.L. (1997). Support for cooperatively controlled objects in multimedia applications. In *Proceedings of CHI'97, Extended Abstracts,* (pp: 313-314). Atlanta: ACM Press.

Clark, H. (1996). *Using Language*. Cambridge University Press, Cambridge.

Czerwinski, M., Horvitz, E. and Cutrell, E. (2001). Subjective duration assessment: An implicit probe for software usability. In *Proceedings of IHM-HCI 2001 Conference, Volume 2, (September, 2001, Lille, France)*, (pp: 167-170).

Dix, A., Finlay, J. Abowd, G. and Beale, R. (1998). *Human-Computer Interaction (2nd Edition)*, Prentice Hall.

Dourish, P. (2001). *Where The Action Is:The Foundations of Embodied Interaction*. Cambridge, MA: MIT Press.

Druin, A., Stewart, J., Proft, D., Bederson, B., and Hollan, J. (1997). KidPad: A design collaboration between children, technologists, and educators. *Proceedings of CHI '97* (pp: 463-470). New York, NY: ACM.

Duncan. S. (1972). Some signals and rules for taking speaking turns in conversations, *Journal of Personality and Social Psychology*, 10, (pp: 283-292).

Dyck, J., and Gutwin, C. (2002). Groupspace: A 3d workspace supporting user awareness. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02)*, (pp: 502-503).

Engelbart, D. and English, W. (1968). A research center for augmenting human intellect. In *Proceedings of the Fall Joint Computing* Conference, 33, (pp: 395-410).

Everitt, K. M., Klemmer, S. R., Lee, R., and Landay, J. A. (2003). Two worlds apart: Bridging the gap between physical and virtual media for distributed design collaboration. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI '03)*, (pp: 553-560).

Finn, K., Sellen, A. and Wilbur, S. (1997). *Video-Mediated Communication*. Lawerence Erlbaum Associates, Inc.

Gerhard, M., Moore, D., and Hobbs, D. (2001). Continuous presence in collaborative virtual environments: towards evaluation of a hybrid avatar-agen model for user representation. In *Proceedings of t he International Conference on Intelligent Virtual Agents (IVA 2001)*, (pp: 137-155).

Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*, Academic Press.

Greenberg, S. and Bohnet, R. (1991). GroupSketch: A multi-user sketchpad for geographically-distributed small groups. In *Proceedings of Graphics Interface '91* (pp: 207-215). Calgary, AB: Morgan-Kaufmann.

Greenberg, S. and Fitchett, C. (2001). Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST 2001*), (pp: 209-218).

Greenberg, S., Gutwin, C., and Roseman, M. (1996). Semantic telepointers for groupware. *Proceedings of the OzCHI '96 Sixth Australian Conference on Computer-Human Interaction* (pp: 54-61). Hamilton, NZ: IEEE Computer Society Press.

Greenberg S. and Roseman, M. (2003). Using a room metaphor to ease transitions in groupware. In M. Ackerman, V. Pipek, V. Wulf (Eds), *Sharing expertise: Beyond knowledge management* (pp: 203-256). Cambridge, MA: MIT Press.

Gross, M., Lang, S., Strenhlke, Moere, A. V., Staadt, O., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-Meier, E., Svoboda, T. and Van Gool, L. Blue-c: a spatially immersive display and 3d video porrtal for telepresence. *ACM Transactions on Graphics*, 22 (3), (pp: 819-827).

Gutwin, C. (1997). *Workspace Awareness in Real-Time Distributed Groupware*. Ph.D Thesis, Dept of Computer Science, University of Calgary, Canada.

Gutwin, C., and Greenberg, S. (2002). A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work*, 11(3-4), (pp: 411-446), Kluwer.

Gutwin, C. and Greenberg, S. (1998). Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proceedings of CSCW'98*, (pp: 207-216).

Gutwin, C., and Penner, R. (2002). Improving interpretation of remote gestures with telepointer traces. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, (pp: 49-57).

Harrison, S. and Minneman, S. (1994). A bike in hand: A study of 3-D objects in design. In Dorst, K, Christiaans, H. and Cross, N. (Eds), *The Delf protocols workshop: Analyzing design activity*, (pp: 205-218).

Ishii, H. (1993). Seamless media design. NTT Interfaces Laboratory, NTT Video. Duration: 5:52.

Ishii, H. (1990). TeamWorkStation: Toward a seamless shared workspace. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '90)*, (pp: 13-26).

Ishii, H. and Kobyashi, M. (1993). Integration of interpersonal space and shared workspace: Clearboard design and experiments. *ACM Transactions on Information Systems*, 11 (4), (pp: 349-375).

Ishii, H., and Kobayashi, M. (1992). ClearBoard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of CHI '92* (pp: 525-532). Monterey: ACM Press.

Ishii, Y., Nakanishi, Y., Koike, H., Oka, K., & Sato, Y. (2004). EnhancedMovie: Movie editing on an augmented desk. In *Adjust Proceedings of the Fifth International Conference on Ubiquitous Computing (Ubicomp 2003), Oct 12-15*. Seattle, WA.

Krauss, R., Dushay, R., Chen, Y. and Rauscher, F. (1995). The communicative value of conversational hand gestures. *Journal of Experimental Social Psychology*, 31, (pp: 533-552).

Kruger, R., Carpendale, M.S.T.. Scott, S. and Greenberg, S. (In press). Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. *Journal of Computer Supported Cooperative Work,* Kluwer Press.

Kruger, R., Carpendale, M.S.T, Tang, A., and Scott, S.D. (2004). Fluid Orientation on a Tabletop Display: Integrating Rotation and Translation. Report 2004-747-12, Department of Computer Science, University of Calgary, Calgary, Canada T2N 1N4, March.

Lombard, M., and Ditton, T. (1997). At the heart of it all: The concept of presence. *Journal of Computer Mediated Communication*, 3(2).

Minneman, S., and Bly, S. (1991). Managing a trois: a study of a multi-user drawing tool in distributed design work. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI '91),* (pp: 217-224).

Miwa, Y., and Ishibiki, C. (2004). Shadow communication: System for embodied interaction with remote partners. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '04),* (pp: 467-476).

Morrel-Samuels, P. and Krauss, R.M. (1992). Word familiarity predicts the temporal asynchrony of hand gestures and speech. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18, (pp: 615-623).

Neustaedter, C., Elliot, K., Tang, A., and Greenberg, S. (2004). Where are you and when are you coming home? Foundations of interpersonal awareness. Technical Report 2004-760-25, Department of Computer Science, University of Calgary, Calgary, Alberta CANADA T2N 1N4.

Pinelle, D., Gutwin, C. and Greenberg, S. (2003). Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Transactions on Human Computer Interaction*, 10(4), (pp: 281-311).

Olson, G. M. and Olson, J. S. (2000). Distance matters. *Human-Computer Interaction*, 15:2, (pp: 139-178).

Ramduny, D., Dix, A., and Rodden, T. (1998). Exploring the design space for notification servers. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'98),* (pp: 227-235). Seattle: ACM Press.

Rauscher, F. B., Krauss, R. M., and Chen, Y. (1996). Gesture, speech and lexical access: The role of lexical movements in speech production. *Psychological Science*, 7, (pp: 226-231).

Riseborough, M. G. (1981). Physiographic gestures as decoding facilitators: Three experiments exploring a neglected facet of communication. *Journal of Nonverbal Behavior*, 5, (pp: 172-183).

Robertson, T. (1997). Cooperative work and lived cognition: A taxonomy of embodied actions. In *Proceedings of the 5th European Conference on Computer Supported Cooperative Work (ECSCW '97)*, (pp: 205-220).

Roseman, M., and Greenberg, S. (1996). Building real time groupware with GroupKit. *ACM Transactions on Computer Human Interaction*, 3(1), 66-106.

Roussel, N. (2001). Exploring new uses of video with VideoSpace. In *Proceedings of the 8th IFIP International Conference on Engeineering for Human-Computer Interaction (EHCI'01)*, LNCS 2254, (pp: 73-90).

Scott, S.D., Carpendale, M.S.T, and Inkpen, K.M. (2004). Territoriality in collaborative tabletop workspaces. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'04)*, November 6-10, 2004, Chicago, IL, USA.

Segal, L. D. (1995). Designing team workstation: The choreography of teamwork. In Hancock, P., Flach, J., Caird, J., and Vicente, K. (eds.), *Local applications of the Ecological Approach to Human-Machine Systems*. Vol 2. New Jersey: Lawrence Erlbaum Associates, Inc.

Segall, B., and Arnold, D. (1997). Elvin has left the building: A publish/subscribe notification service with quenching. In *Proceedings of AUUG '97*, Brisbane, Australia.

Sellen, A., Buxton, W. and Arnott, J. (1992). Using spatial cues to improve videoconferencing. In *Proceedings of ACM Conference on Human-Computer Interaction (CHI '92)*, (pp: 651-652).

Short, J., Williams, E., and Christie, B. (1976). Communication modes and task performance. In Baecker, R. M. (ed.), *Readings in Groupware and Computer Supported Cooperative Work*, (pp: 169-176). Mountain View, CA: Morgan-Kaufman Publishers.

Starner, T., and Pentland, A. (1995). Visual recognition of american sign language using hidden markov models, In *Proceedings of the International Workshop on Face and Gesture Recognition*.

Takahashi, T., and Kishino, F. (1991). Hand gesture coding based on experiemnts using a hand gesture interface device. *SIGCHI Bulletin*, 22(2), (pp: 67-73).

Tang, A., Boyle, M. and Greenberg, S. (2004). Display and presence disparity in mixed presence groupware. In *Proceedings of the 5th Australasian User Interface Conference (AUIC'04)*, (pp: 73-82).

Tang, J. (1991). Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2), 143-160.

Tang, J. & Minneman, S. (1991a). VideoDraw: A video interface for collaborative drawing. *ACM Transactions on Information Systems*, 9 (2), April 1991.

Tang, J. & Minneman, S. (1991b). VideoWhiteboard: Video shadows to support remote collaboration. In *Proceedings of CHI '91* (pp: 315-32). New Orleans: ACM Press.

Tse, E., and Greenberg, S. (2002). SDGToolkit: A toolkit for rapidly prototyping single display groupware. In *Extended Abstracts of CSCW 2002*, (pp: 173-174). New Orleans: ACM Press.

Tse, E., Histon, J., Scott, S. D., and Greenberg, S. (2004). Avoiding interference: How people use spatial separation and partitioning in SDG workspaces. In *Proceedings of the ACM CSCW '04 Conference on Computer Supported Cooperative Work*, (Nov 6-10, Chicago, Illinois), ACM Press.

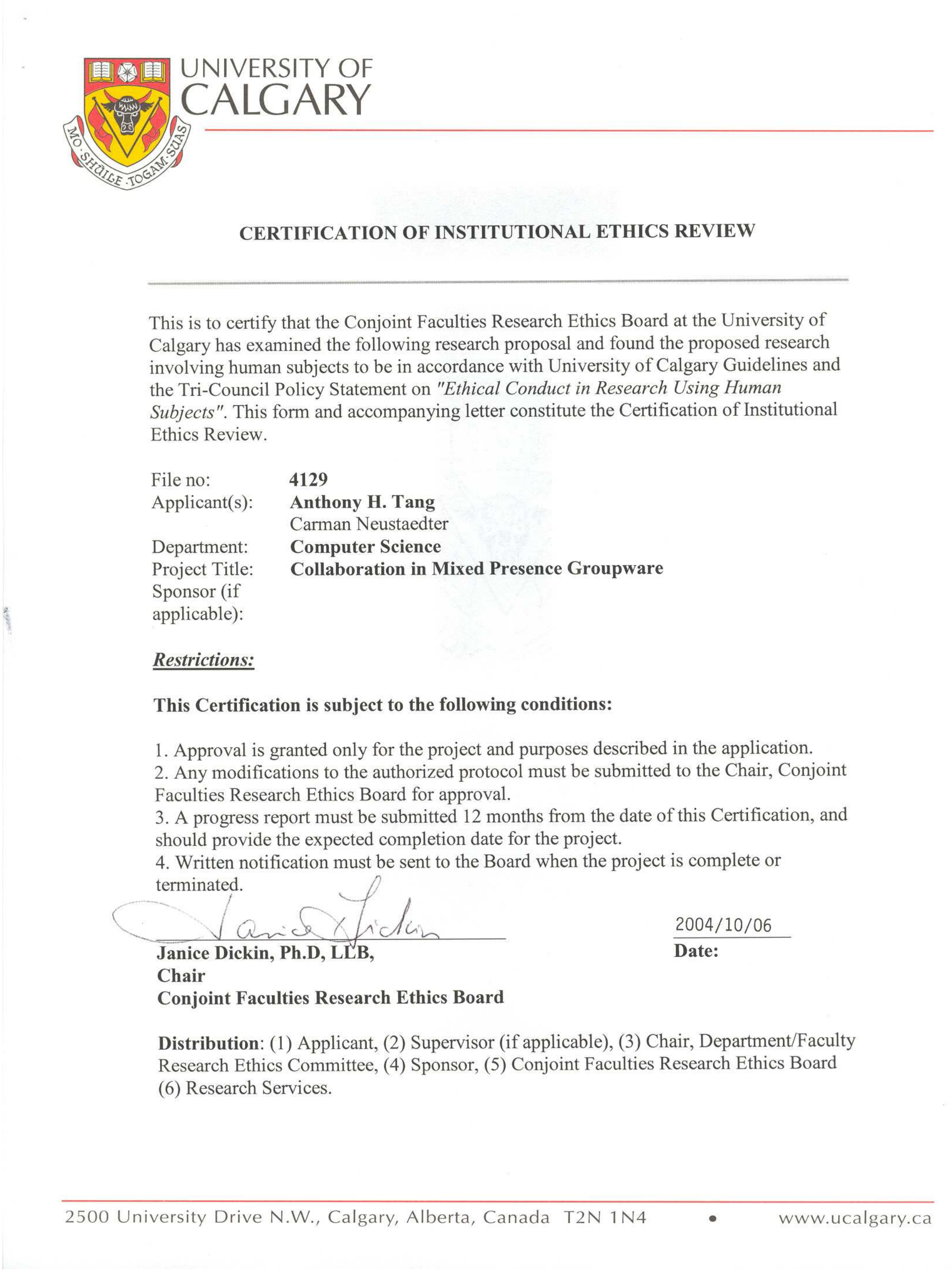
Vertegaal, R. (1999). The GAZE groupware system: Mediating joint attention in multiparty communication and collaboration. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI '99)*, (pp: 294-301).

Vertegaal, R. Weevers, I. and C. Sohn. (2002). GAZE-2: An attentive video conferencing system. In *Extended Abstracts of ACM Conference on Human-Computer Interaction (CHI '02)*, (pp: 736-737).

Xerox PARC. (1987). *The Office Design Project*. Systems Concept Laboratory, Video report.

Zanella, A., and Greenberg S. (2001). Reducing interference in singlde display groupware through transparency. In *Proceedings of the Sixth European Conference on Computer Supported Work (ECSCW '01)*, (pp: 339-358).

Appendix A. Ethics Approval



UNIVERSITY OF
CALGARY

CERTIFICATION OF INSTITUTIONAL ETHICS REVIEW

This is to certify that the Conjoint Faculties Research Ethics Board at the University of Calgary has examined the following research proposal and found the proposed research involving human subjects to be in accordance with University of Calgary Guidelines and the Tri-Council Policy Statement on *"Ethical Conduct in Research Using Human Subjects"*. This form and accompanying letter constitute the Certification of Institutional Ethics Review.

File no:	**4129**
Applicant(s):	**Anthony H. Tang** Carman Neustaedter
Department:	**Computer Science**
Project Title:	**Collaboration in Mixed Presence Groupware**
Sponsor (if applicable):	

Restrictions:

This Certification is subject to the following conditions:

1. Approval is granted only for the project and purposes described in the application.
2. Any modifications to the authorized protocol must be submitted to the Chair, Conjoint Faculties Research Ethics Board for approval.
3. A progress report must be submitted 12 months from the date of this Certification, and should provide the expected completion date for the project.
4. Written notification must be sent to the Board when the project is complete or terminated.

Janice Dickin

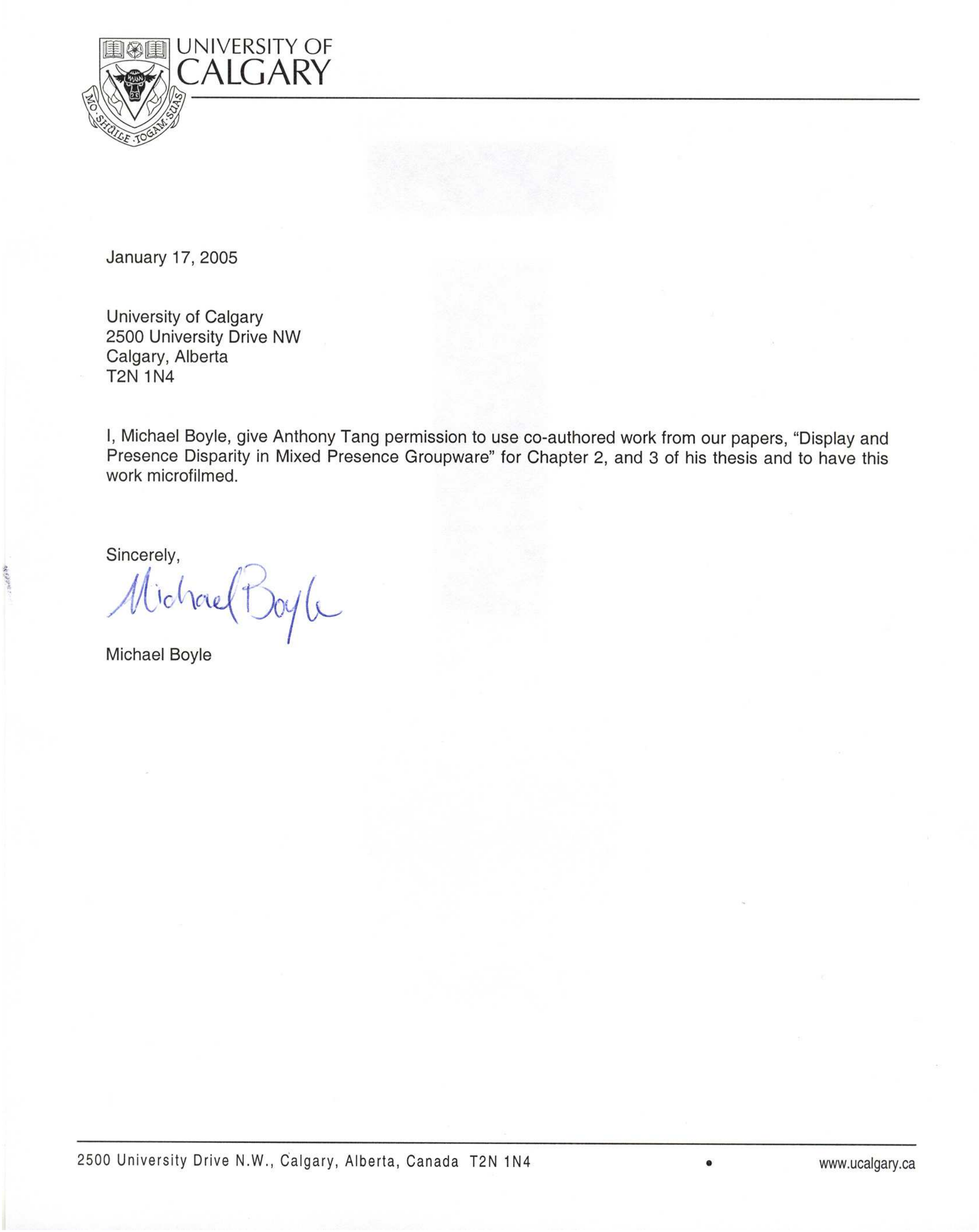
Janice Dickin, Ph.D, LLB,
Chair
Conjoint Faculties Research Ethics Board

2004/10/06
Date:

Distribution: (1) Applicant, (2) Supervisor (if applicable), (3) Chair, Department/Faculty Research Ethics Committee, (4) Sponsor, (5) Conjoint Faculties Research Ethics Board (6) Research Services.

2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 • www.ucalgary.ca

Appendix B. Co-Author Permission



UNIVERSITY OF
CALGARY

January 17, 2005

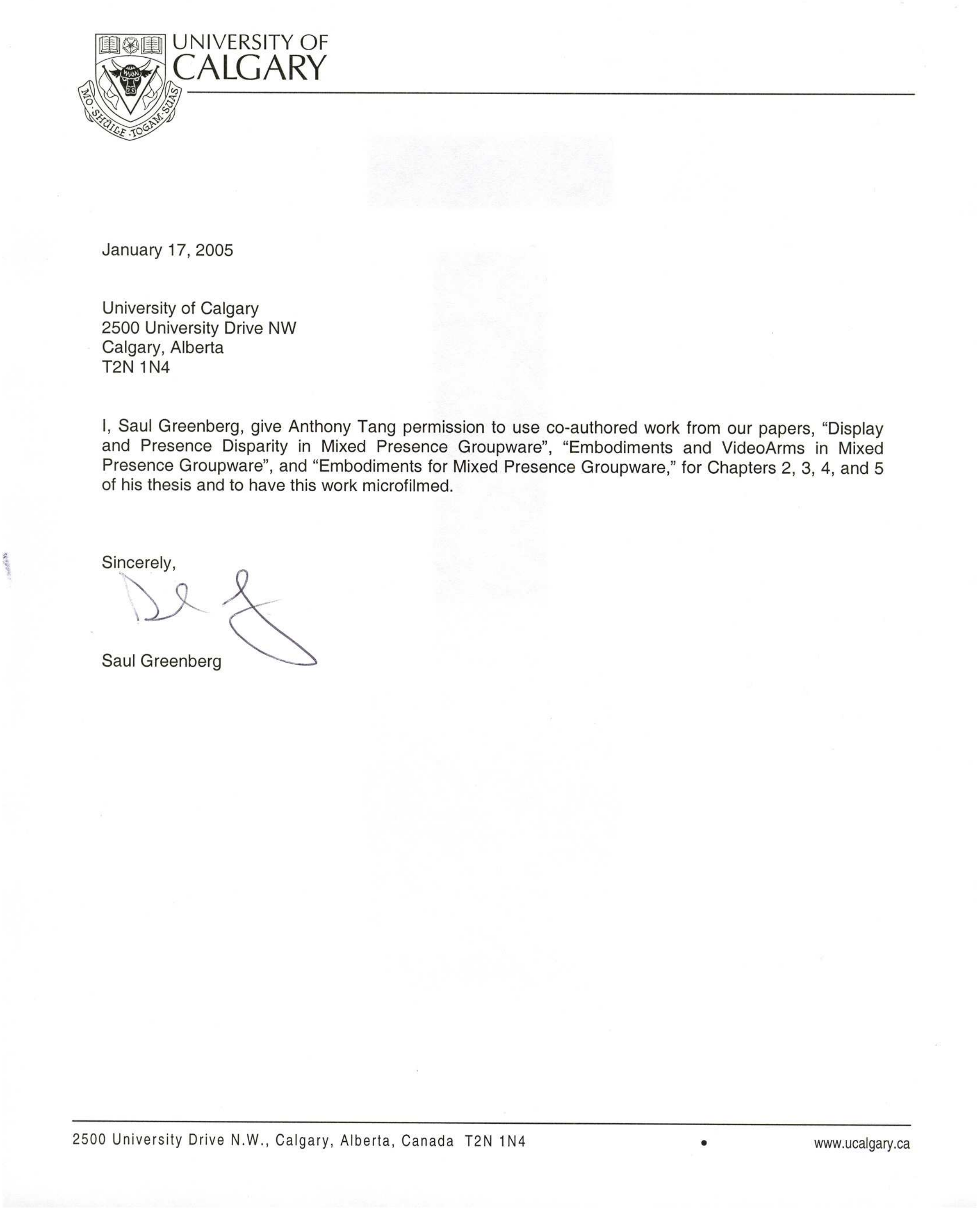
University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

I, Michael Boyle, give Anthony Tang permission to use co-authored work from our papers, “Display and Presence Disparity in Mixed Presence Groupware” for Chapter 2, and 3 of his thesis and to have this work microfilmed.

Sincerely,

Michael Boyle

Michael Boyle



UNIVERSITY OF
CALGARY

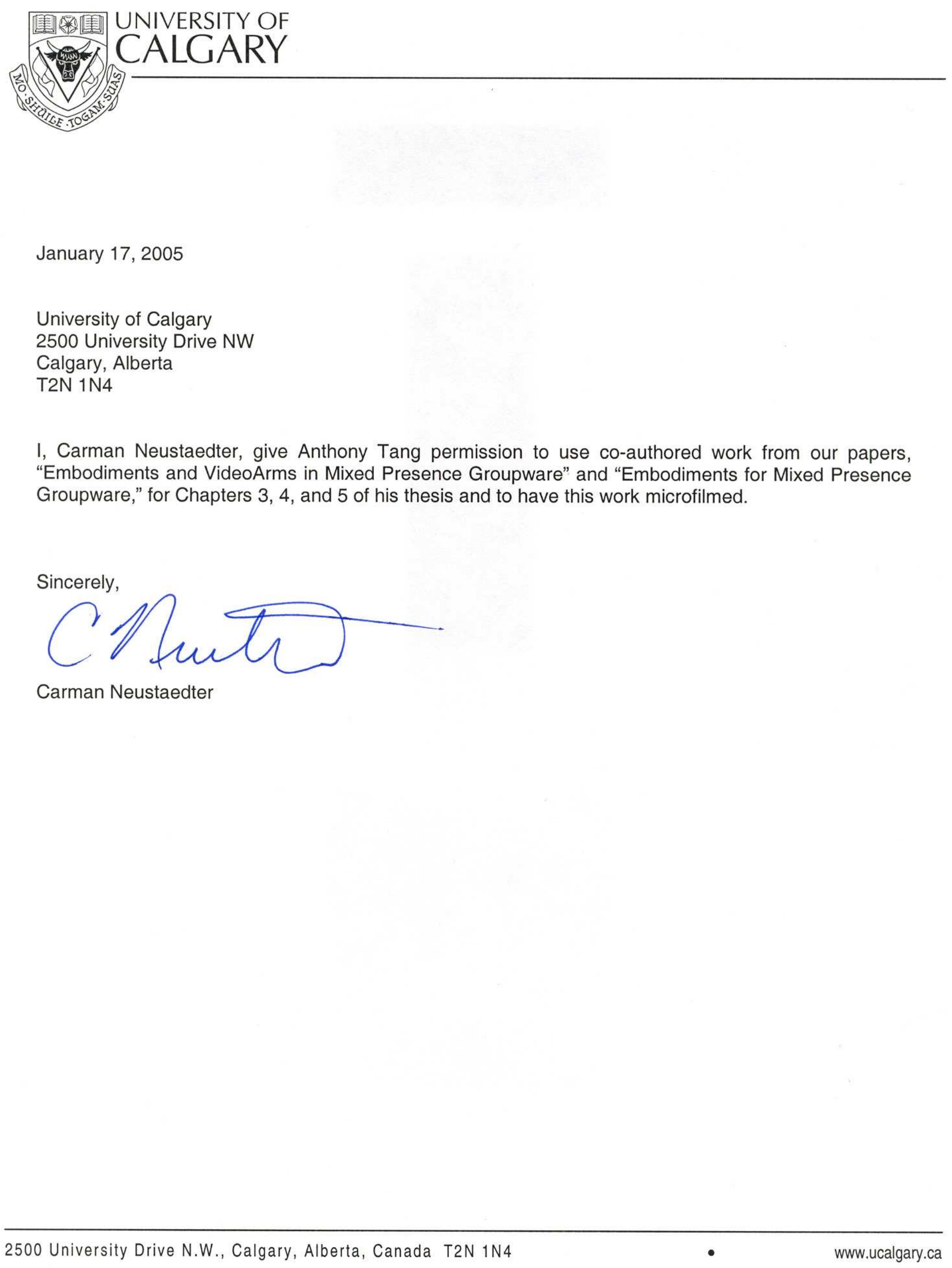
January 17, 2005

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

I, Saul Greenberg, give Anthony Tang permission to use co-authored work from our papers, “Display and Presence Disparity in Mixed Presence Groupware”, “Embodiments and VideoArms in Mixed Presence Groupware”, and “Embodiments for Mixed Presence Groupware,” for Chapters 2, 3, 4, and 5 of his thesis and to have this work microfilmed.

Sincerely,

Saul Greenberg



UNIVERSITY OF
CALGARY

January 17, 2005

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

I, Carman Neustaedter, give Anthony Tang permission to use co-authored work from our papers, “Embodiments and VideoArms in Mixed Presence Groupware” and “Embodiments for Mixed Presence Groupware,” for Chapters 3, 4, and 5 of his thesis and to have this work microfilmed.

Sincerely,

Carman Neustaedter