Tangible Interfaces for Remote Collaboration and Communication

Scott Brave, Hiroshi Ishii, and Andrew Dahley
MIT Media Laboratory
Tangible Media Group
20 Ames St., Cambridge, MA 02139
{brave,ishii,andyd}@media.mit.edu

ABSTRACT

Current systems for real-time distributed CSCW are largely rooted in traditional GUI-based groupware and voice/video conferencing methodologies. In these approaches, interactions are limited to visual and auditory media, and shared environments are confined to the digital world. This paper presents a new approach to enhance remote collaboration and communication, based on the idea of Tangible Interfaces, which places a greater emphasis on touch and physicality. The approach is grounded in a concept called Synchronized Distributed Physical Objects, which employs telemanipulation technology to create the illusion that distant users are interacting with shared physical objects. We describe two applications of this approach: PSyBench, a physical shared workspace, and inTouch, a device for haptic interpersonal communication.

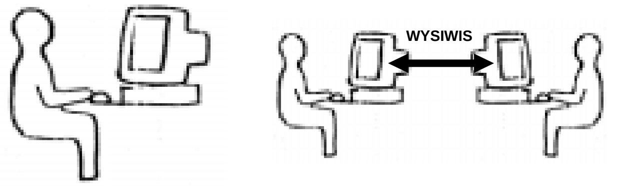
Keywords

Tangible Interfaces, haptic interfaces, telemanipulation, force-feedback, physical presence

INTRODUCTION

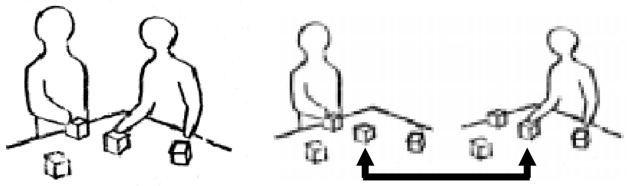
For many years our conception of human-computer interaction has been focused on the Graphical User Interface (GUI) (Figure 1a). GUIs allow interaction with digital objects and online information through the generic screen, keyboard, and pointing device. Current systems for Computer Supported Cooperative Work (CSCW) are largely based on extensions of the GUI to a distributed multi-user context, providing distant users with shared access to online digital environments (Figure 1b). When direct communication between distributed users is desired, these systems are traditionally augmented with voice/video conferencing technologies.

In the real world, touch and physical manipulation play a key role in understanding and affecting our environment [12]. Traditional interfaces to the digital world, in contrast, largely fail to address our sense of touch and offer only the generic keyboard and pointing device as tools for indirect manipulation of digital objects.



a) Graphical User Interface (GUI)

b) Real-time, distributed CSCW based on GUI



c) Tangible User Interface (TUI)

d) Real-time, distributed CSCW based on TUI

Figure 1. Interface Techniques for HCI and CSCW.

Physicality also plays an important role in interpersonal communication (consider the impact of a strong handshake or a nudge for attention). However, current GUI-based systems for distributed interactions, provide no means for this type of physical communication or awareness.

We have previously introduced Tangible User Interfaces (TUIs) as an alternative to the GUI that makes greater use of physical space and real-world objects as interface tools (Figure 1c) [8]. This paper presents an approach called Synchronized Distributed Physical Objects which enables the extension of Tangible User Interfaces into the space of distributed CSCW (Figure 1d). The goal is to enhance real-time remote collaboration and communication by bringing a greater sense of touch and physicality to distributed multi-user interactions. We describe two applications of this approach. PSyBench provides a generic shared physical workspace for distributed users. We present an early prototype of PSyBench, built from two motorized chessboards, and discuss relevant interface issues. We then present inTouch, which applies Synchronized Distributed

Physical Objects to create a "tangible telephone" for long-distance haptic communication.

Tangible Interfaces

Tangible Interfaces [8] represents a general approach to human-computer interaction that puts greater emphasis on physicality than traditional graphics-based interfaces. Illuminating Light [20] is one example of a Tangible Interface for optical design and layout (Figure 2). In this system, users directly arrange and manipulate physical objects representing lasers, mirrors, lenses, and other optical components on an augmented tabletop. The positions of these objects are recognized by the system and the behavior of the laser light is projected onto the table in the same physical space as the optical components. Users are thus able to make full use their hands and bodies in affecting the simulation, as well as use their spatial and kinesthetic senses in understanding the arrangements. Other examples of Tangible Interfaces include the metaDESK [18], Triangles [7], and mediaBlocks [19].

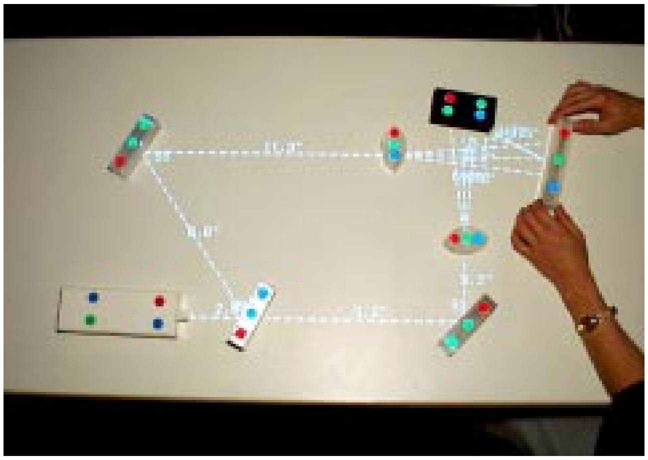


Figure 2. Illuminating Light, a Luminous-Tangible Interface for holography simulation.

A big advantage of Tangible Interfaces is that they support multi-user interactions well. Since a generic pointing device is not needed to mediate interactions, many users can interact with a Tangible Interface system in parallel. In Illuminating Light, for example, multiple users can simultaneously grab and manipulate the optical components to cooperatively create and explore simulated holography layouts.

An important next question is, how can such an object-based interface be used in a distributed context? One solution would simply be to give each *separate* space their own interface objects and then project a video capture of remote spaces onto the local setup, in a way similar to TeamWorkStation [10] (also see [13]). This may be unsatisfactory, however, as local users may want to manipulate objects in remote spaces, as well as in their own local space. Synchronized Distributed Physical Objects presents an approach that allows distant users to share physical objects across distance, enabling the extension of Tangible Interfaces into the space of distributed multi-user interactions.

SYNCHRONIZED DISTRIBUTED PHYSICAL OBJECTS

Imagine that you are an urban planner and are trying to design the layout of a new college campus with a remote colleague. You sit down at a table and place on it a blueprint of the area and a number of scaled models representing each of the landmarks you wish to arrange. Your remote colleague has the same blueprint and set of models and places them on her table. Using both hands, you begin to arrange the physical building models in the central campus. At the same time you are positioning and adjusting the central campus, you see the physical models representing the laboratory clusters moving around on your table in the are designated “east campus”. Recognizing your remote colleague's struggle with fitting in all the lab buildings, from her frequent subtle yet unsuccessful tweaks, you grab two of the lab buildings and suggest a new arrangement by moving them to the other side of the campus. On her table, she sees the models move as you make the suggestion and then begins to move her gaze around the table space to get a few different views of the area and your changes.

The above scenario is representative of the Synchronized Distributed Physical Objects vision. Traditional CSCW systems have long allowed distributed users to share *digital* objects and environments (Figure 3a). Synchronized Distributed Physical Objects allow distant users to share *physical* objects and environments as well (Figure 3b).

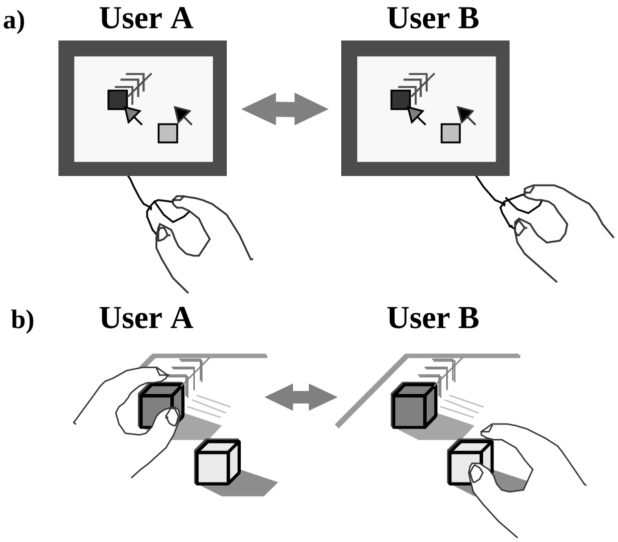


Figure 3. Distributed shared spaces. a) A shared digital space. b) A shared physical space.

A Synchronized Distributed Physical Object creates the illusion of a shared physical object across distance by physically synchronizing the states of distant, identical copies of an object, using telemanipulation technology.

Sensors (e.g. optical encoders, cameras) monitor the states of the distributed copies of a "shared" object and actuators (e.g. motors) are employed to synchronize those states. Thus, when a local user manipulates her local copy of a shared physical object, she is effectively manipulating all remote copies as well (Figure 3b). Distributed users can then share physical objects, able to both manipulate and see others' manipulation of the same objects.

Level of Synchronization

The level to which physical synchronization is implemented can be dictated by the dynamics of the intended application. Although ideally we would like the ability to tightly synchronize all aspects of a shared object (including 3D physical location and internal state), this is often not technically feasible or worthwhile. Depending on the application, it may be adequate--and perhaps even preferable--to synchronize only some physical aspects of the shared objects or relax the synchronization to a looser coupling.

PSyBench, for example, is intended as a generic platform for extending shared physical workspaces, such as Illuminating Light, into a distributed multi-user context. For these types of applications (the urban planning scenario described above is another example), synchronizing the 2D positions and orientations of objects only when they are on an augmented tabletop is reasonable. PSyBench also temporarily suspended synchronization if multiple users move the same objects at once. inTouch, on the other hand, a device for haptic interpersonal communication, exploits a much tighter coupling that maintains synchronization when multiple users simultaneously manipulate an object (Figure 4); however, synchronization is limited to one degree-of-freedom of three cylindrical rollers embedded with a base. This tight coupling provides a channel for direct physical communication between distant users.

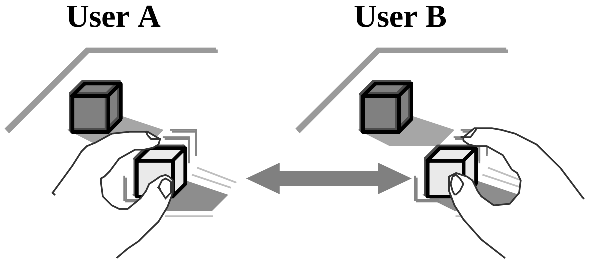


Figure 4. Synchronization of a shared physical object being simultaneously manipulated by multiple users.

PSYBENCH

PSyBench (Physically Synchronized Bench) employs the concept of Synchronized Distributed Physical Objects to provide a generic shared physical workspace across distance. The goal is to allow distributed users to cooperate in Tangible Interface applications, such as Illuminating Light, which are heavily based around physical objects. To do this, we turn each physical interface object into a Synchronized Distributed Physical Object so that it can be shared by distant users.

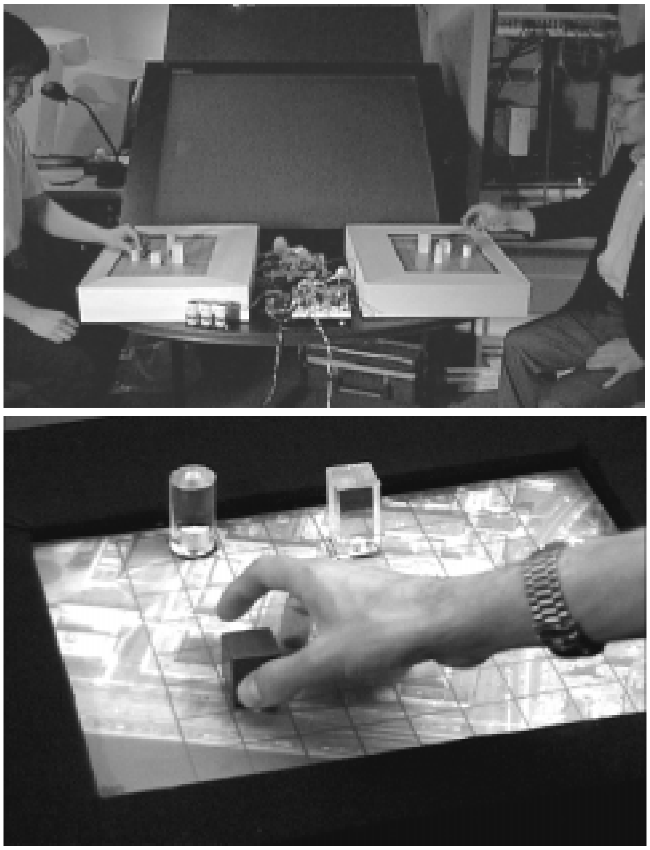


Figure 5. Early prototype of PSyBench.

An initial prototype of PSyBench is constructed from two augmented and connected motorized chessboards from Excalibur (Figure 5). Positions of objects on a ten-by-eight grid are sensed by an array of membrane switches. The objects have magnetic bases so that they can be moved using an electromagnet placed on a 2-axis positioning mechanism under the surface. Each board is outfitted with custom hardware, based around the PIC microprocessor, to handle the control and serial communication between boards. Figure 6 shows this system architecture.

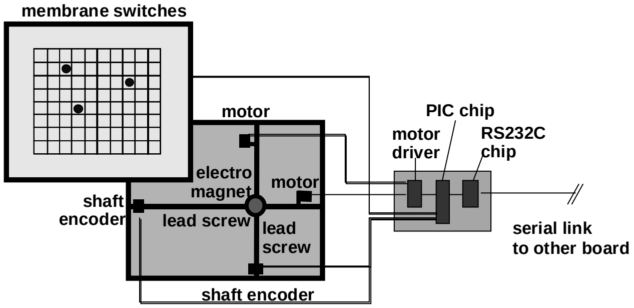


Figure 6. System architecture for PSyBench prototype.

This early prototype has obvious limitations; most notably, positioning is discrete and there is no mechanism for synchronizing the orientation of objects. However, the system has been extremely helpful in bringing to light many implementation issues, as well as design implications.

Tangible Presence

PSyBench primarily provides a means for geographically distant users to collaborate in a shared physical workspace, extending the benefits of Tangible Interfaces into a distributed CSCW context. Initial experiences with the prototype system, however, have suggested that PSyBench also presents a new form of "awareness" of the *physical* presence of remote collaborators. The actions of remote users are manifested in a physical and tangible way, as motion of grasped objects, that suggests form and movement of a motivating physical body. Much in the way that a player piano compels us to imagine a real body sitting at the piano bench with arms extending to the keys, the movement of objects on PSyBench, many initial users have commented, evokes strong feelings of physical presence.

This feeling is particularly compelling considering that the objects affected by remote users are not in some distant or removed space, but in the same space you yourself are sitting and acting in. Objects manipulated by distant users are the same objects that you can touch and feel with your hands; they may even get in your way or touch you as they move. In this way, the shared workspace of objects and your physical interpersonal space are seamlessly integrated, much in the way that ClearBoard integrates the two on a visual level [9].

Co-located Vs. Distributed Users

Another important property of PSyBench is that all users, be they local or remote, essentially interact with the workspace in the same way: by manipulating physical objects on the augmented table's surface. One major difference in person-to-person interactions between co-located and distributed users, however, is the lack of direct visual presence of remote users. Co-located users, for example, are able to see one another's hands moving toward an object before they begin manipulating it. This cue can provide an early means for avoiding the situation where two users try to grab the same object simultaneously. A similar situation arises if a user who has been moving an object pauses to think but still keeps his/her hand on object. In the described system, co-located users would be able to see this behavior, while distributed users would lack this visual cue.

We are currently experimenting with several options to address these issues including projecting a direct visual overlay of remote users' hands onto the table [13] and projecting more abstract representations of users' hands and/or object contact. Integration of more traditional video-conferencing techniques through table projection is also a possibility. One particularly interesting setup could be created if a wall was placed abutting one side of the table onto which a remote user's space was projected in a way similar to ClearBoard [9], providing both direct and task-oriented "gaze awareness" for distributed users.

INTOUCH

Touch is often recognized as a fundamental aspect of interpersonal communication. Whether a strong handshake, an encouraging pat on the back, a nudge for attention, or a gentle brush of a shoulder, physical contact can convey a vitality and immediacy at times more powerful than language. Touch can instantly indicate the nature of a relationship; it is sincere, immediate, and compelling. Yet while many traditional technologies allow communication through sound or image, none are designed for expression through touch. inTouch is a system for haptic interpersonal communication based on the concept of Synchronized Distributed Physical Objects.

Previous work in Telehaptic Communication

Although sparse, there have been a few projects that explore haptic interpersonal communication (or telehaptic communication). *Telephonic Arm Wrestling* [21] provides a basic mechanism to simulate the feeling of arm wresting over a telephone line. *Denta Dentata* [6] is an elementary "hand holding" device that communicates one bit of information over the phone line to activate a mechanism that can squeeze a user's hand. *Feather, Scent, and Shaker* [17] consists of a pair of linked “shaker” objects. Shaking one object causes the other to vibrate, and vice-versa. *HandJive* [5] is a pair of linked hand-held objects for playing haptic games. Each object has a joystick-like controller that can be moved vertically or horizontally. A horizontal displacement of the local object causes a vertical displacement in the remote object, and vice-versa. *Kinesthetic Constructions* [15] explores the application bilateral force-feedback to interpersonal communication. Schena describes a network of large modern sculptures distributed around the world where parts of each sculpture are hapticly connected to sculptures at other locations.

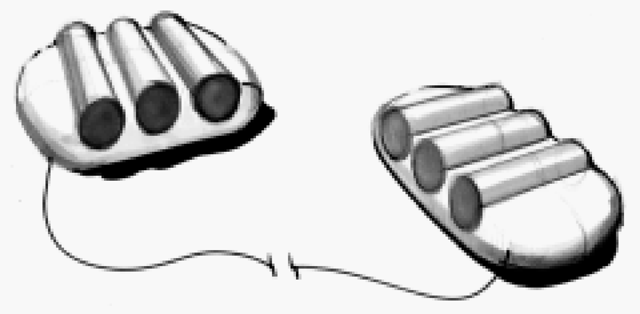


Figure 7. inTouch concept sketch.

inTouch Design

In the spirit of many of these explorations, inTouch provides a system for haptic interpersonal communication across distance. Of these other works, inTouch perhaps resembles the idea of *Kinesthetic Constructions* most closely in that the interaction is bilateral (fully integrated 2-way) and general (without a designated task space or connotation). As seen in Figure 7, inTouch consists of two hand-sized objects each with three cylindrical rollers embedded within a base. Employing the Synchronized Distributed Physical Objects concept, the rollers on each base are hapticly coupled such that each one feels like it is physically linked to its counterpart on the other base. To achieve the tight coupling necessary to allow simultaneous manipulation, inTouch employs bilateral force-feedback technology, with position sensors to monitor the states of the rollers and high precision motors to synchronize those states. Two people separated by distance can then passively feel the other person's manipulation of the rollers, cooperatively move the "shared" rollers, or fight over the state of the rollers, providing a means for expression through touch.

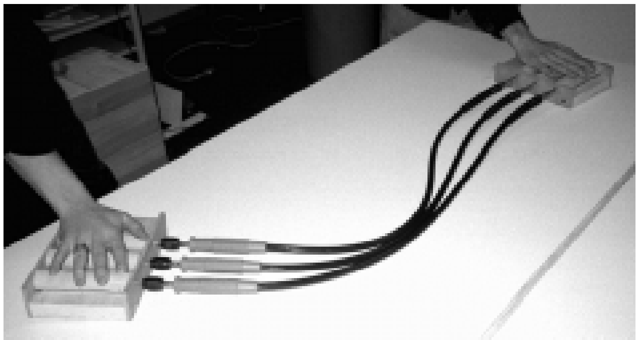


Figure 8. Mechanical mockup of inTouch (inTouch-0). Corresponding rollers are connected using flexible drive shafts.

Mechanical Mockup: inTouch-0

Figure 8 shows an early mockup of inTouch where corresponding rollers were actually mechanically connected using flexible drive shafts (see [1] for a discussion of this mockup as well as inTouch design decisions). This model was implemented in a graduate course on interface design, in October 1996, and was presented in class. Users often described the interaction as fun or playful, with one student relating the experience to when he and his sister would use a broom to play tug-of-war as children. Some remarked that the lack of ability to pass concrete information made the medium uninteresting, while others applauded the subtle and abstract nature of the interaction. This mechanical mockup can been seen as a benchmark for creating the distributed version, since it is this feeling of direct connection that we are aiming to simulate across distance.

Figure 9. Prototype of inTouch where corresponding rollers are connected virtually, using force-feedback technology.

Standalone Prototype: inTouch-1

InTouch-1 was created next to implement the connection between rollers, virtually, using force-feedback technology (Figure 9). Ideally, the goal is to have virtually connected rollers that behave identically to the mechanically connected rollers in inTouch-0.

The system architecture for inTouch-1 is shown in Figure 10. Hewlett Packard optical position encoders were used to monitor the physical states of the rollers (positions were read directly, other values were interpolated) and high performance Maxon DC motors were used to synchronize those states. A 200MHz Pentium PC controlled all motor/encoder units (one unit for each roller) using Immersion Corporation's Impulse Drive Board 1.0 boards and 2-Axis Card 1.0 ISA cards.

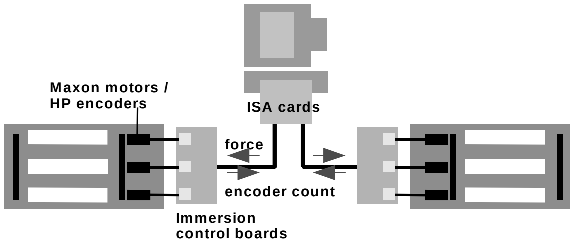


Figure 10. inTouch-1 system architecture (standalone prototype).

The control algorithm that ran on the host PC simulates a highly damped, stiff rotary spring between corresponding rollers. In other words, the algorithm looks at the difference in position of each pair of "connected" rollers and applies a restoring force, proportional to that difference, to bring the rollers together (see the Appendix for an in depth discussion of the control algorithm and optimization).

The first prototype of inTouch-1 was completed in March 1997, and has been demonstrated at sponsor meetings and at the 1997 Ars Electronica Festival, as well as tested internally. People who knew the previous version, inTouch-0, were surprised at how closely the interaction matched the mechanical mockup. In total, more than 500 people have tried inTouch, several of whom have made enthusiastic requests for the system to "keep in touch" with distant family and loved ones. Many people have indicated their belief that inTouch provides a means to be aware of a distant person's emotional state and sincerity, however, we have not yet formally tested this proposition.

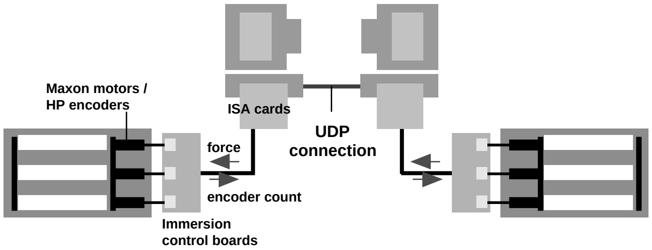


Figure 11. inTouch-2 system architecture.

Networked Prototype: inTouch-2

Our current prototype, inTouch-2, allows the virtual connection of inTouch-1 to be extended over arbitrary distance, using the Internet.

The system architecture for inTouch-2 is shown in Figure 11. The architecture is identical to that of inTouch-1 except that the two sets of three rollers run on separate host computers, distributed over a standard network. Positions and velocities of the local rollers are passed to the remote computer using User Datagram Protocol (UDP).

The basic control algorithm for the networked design is also the same as that for inTouch-1. Each computer simply calculates the forces to impart to its three rollers given the state of each local roller (received from the local control hardware) and the most recently received position and velocity of the corresponding remote roller (passed over the network by the other PC).

We have so far distributed inTouch-2 over the local area network in our building. At this distance, with a little modification to the control algorithm (see Appendix), inTouch-2 behaves identically to inTouch-1. Simulations of longer distances, and consequently longer network delays, have shown promise in extending inTouch over arbitrary distances (see Appendix).

FUTURE WORK

We designed the two prototypes illustrated in this paper, PSyBench and inTouch, as a means to explore a new design space for CSCW and demonstrate the potential of distributed Tangible Interfaces. Although still in the early stages, these explorations have raised a number of interesting and tough research questions. We are now focusing on the following three directions as our future work:

1) Developing robust and extendable platforms for Synchronized Distributed Physical Objects. For example, we are currently developing a larger table-sized version of PSyBench, which detects objects through a combination of machine vision and electromagnetic field sensing, and employs a larger magnetic positioning system for actuation.

2) Identifying collaborative applications which can take full advantage of shared physical objects, coupled with digital augmentation, and

3) Investigating the implications and appropriate applications of a haptic interpersonal communication link through experimentation and long-term user testing.

CONCLUSION

The personal computer has enabled distant users to work together by providing distributed access to shared digital environments. Limited by available interface technology, however, collaborations in these digital spaces often pale in comparison to the richness and facility of interactions in the physical world. In co-located situations, for example, collaborators often rely on the ability to interact with various shared physical objects and appreciate the physical co-presence with others. Traditional interfaces to the digital world, in contrast, tend to impoverish our sense of touch and limit our physical interactions to typing on a generic keyboard or manipulating a plastic mouse.

In this paper, we have introduced a concept called Synchronized Distributed Physical Objects, which poses a new approach to addressing this lack of physicality in GUI-based CSCW interfaces. We have introduced two prototype systems that begin exploration of this new design space for distributed multi-user systems. PSyBench allows distributed users to cooperate in a shared physical workspace, where the presence of remote users is manifested, tangibly, as the movement of shared physical objects. inTouch provides a "tangible telephone" to enable haptic interpersonal communication across distance.

What You See Is What I See (WYSIWIS) has long been a guiding principle for the design of shared *digital* spaces. Synchronized Distributed Physical Objects offer an extension of the WYSIWIS abstraction into the *physical* world. Synchronized Distributed Physical Objects can be seen first as Physical WYSIWIS, since all users will *see* other users' manipulation of the shared physical object. In implementations that use a tight coupling, such as inTouch, What You Feel Is What I Feel will also hold, since all users will be able to simultaneously manipulate and *feel* other users' manipulation of the shared object. As we have mentioned, the idealized notion of strict synchronization may need to be relaxed for technical and/or interface reasons, as is often true with WYSIWIS as well [16]. However, the general principle of Synchronized Distributed Physical Objects can be used as a guide in the design of distributed Tangible Interfaces.

ACKNOWLEDGMENTS

We would like to thank the many fellow Tangible Media Group members who have contributed ideas and time to this project. In particular, we thank Phil Frei for his continued and excellent work in building the mechanical part of inTouch (used in inTouch-1 and -2), Victor Su for his continued efforts in designing the electronics for a next-generation version of inTouch, and Colyn Bulthaup for his hard work in implementing the PSyBench prototype. We would also like to thank Paul Yarin and Wendy Plesniak for their comments on drafts of this paper. This work is funded in part by the National Science Foundation.

REFERENCES

1. Brave, S., and Dahley, D. inTouch: a medium for haptic interpersonal communication. *Extended Abstracts of CHI'97* (Atlanta GA, Mach 1997). ACM Press, 363-364.
2. Burdea, G.C. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, Inc., New York NY, 1996.
3. Buxton, W. The three mirrors of interaction: a holistic approach to user interfaces. *Proceedings of Friend21 '91 International Symposum on Next Generation Human Interface* (Tokyo, Japan, November 1991). 25-27.
4. Buxton, W. Touch, Gesture & Marking. Chapter 7 in R.M. Baecker, J. Grudin, W. Buxton and S. Greenberg, S. (Eds.) *Readings in Human Computer Interaction: Toward the Year 2000.* Morgan Kaufmann Publishers, San Francsco CA, 1995.
5. Fogg, B.J., Cutler, L., Arnold, P., and Eisback, C. HandJive: a device for interpersonal haptic entertainment, in *Proceedings of CHI '98* (Los Angeles CA, April 1998), ACM Press. 57-64.
6. Goldberg, K., and Wallace, R. Denta Dentata, in *Visual Proceedings of SIGGRAPH '93*. ACM Press.
7. Gorbet, M.G., Orth, M., and Ishii, H. Triangles: Tangible Interfaces for manipulation and exploration of digital information topography, in *Proceedings of CHI '98* (Los Angeles CA, April 1998), ACM Press. 49-56.
8. Ishii, H., and Ullmer, B. Tangible Bits: towards seamless interfaces between people, bits and atoms, in *Proceedings of CHI '97* (Atlanta GA, Mach 1997), ACM Press, 234-241.
9. Ishii, H., Kobayashi, M., and Grudin, J. Integration of inter-personal space and shared workspace: ClearBoard design and experiments. in *Proceedings of CSCW '92* (Toronto, November 1992), ACM Press, 33-42.
10. Ishii, H. TeamWorkStation: towards a seamless shared workspace, in *Proceedings of CSCW '90* (Los Angeles CA, October 1990), ACM Press, 13-26.
11. Jackson, B., and Rosenberg, L. Force feedback and medical simulation, in Morgan, K., Satava, R., Sieberg, H., Mattheus, R, and Christensen, J. (Eds.) Interactive Technology and the New Paradigm for Healthcare (Amsterdam, January 1995), IOS Press, 147-151.
12. Johnson, M. *The Body in the Mind*. The University of Chicago Press, Chicago IL, 1987.
13. Krueger, M. K. *Artificial Reality II*. Addison Wesley Publishing, Reading MA, 1991.
14. Massie, T.H., and Salisbury, J.K. The Phantom haptic interface: a device for probing virtual objects, in *Proceedings of the 1994 ASME International Mechanical Engineering Congress and Exhibition* (Chicago IL) VOL DSC 55-1, ASME Press, 295 - 302.
15. Schena, B. Design of a Global Network of Interactive, Force-Feedback Sculpture. Master's Thesis in the Department of Mechanical Engineering, Stanford University, 1995.
16. Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., and Tatar, D. WYSIWIS revised: early experiences with multiuser interfaces, in Baecker, R. M. (Ed.), *Readings in Groupware and Computer-Supported Cooperative Work, Assisting Human-Human Collaboration*. Morgan Kaufmann Publishers, Inc., San Mateo CA, 1993, 585-595.
17. Strong, R., and Gaver, B. Feather, Scent and Shaker: Supporting Simple Intimacy, in *Videos, Demonstrations, and Short Papers of CSCW '96* (Boston MA, November 1996), 29-30.
18. Ullmer, B., and Ishii, H. The metaDESK: models and prototypes for Tangible User Interfaces, in *Proceedings of UIST '97, User Interface Software Technology* (Banff, October 1997), ACM Press. 223-232.
19. Ullmer, B., Ishii, H, and Glass, D. mediaBlocks: physical containers, transports, and controls for online media, in *Proceedings of SIGGRAPH '98* (Orlando FL, July 998), ACM Press.

20. Underkoffler, J., and Ishii, H. Illuminating Light: an optical design tool with a luminous-tangible interface, to appear in *Proceedings of CHI '98* (Los Angeles CA, April 1998), ACM Press. 542-549.

21. White, N., and Back D. Telephonic Arm Wrestling, shown at *The Strategic Arts Initiative Symposium* (Salerno, Italy, Spring 1986). See http://www.bmts.com /~normill/artpage.html

APPENDIX A: CONCEPTUAL FRAMEWORK

This appendix illustrates the conceptual framework addressed in this paper in more detail (Figure 12). Current approaches to Human Computer Interaction are largely based on the Graphical User Interface. GUIs allow interaction with digital objects and environments through the generic screen, mouse, and keyboard. Traditional approaches to CSCW employ "What You See Is What I See" (WYSIWIS) to extend this approach into the area of distributed multi-user interactions, allowing distant users to interact in a shared digital space. When direct communication between users is desired, these systems are often augmented with traditional video/telephony.

Tangible Interfaces [8] provides an alternative to these traditional approaches that moves the focus of interaction off of the screen and into the physical world. The aim is to exploit the richness of the physical world while allowing users to make use of their spatial manipulation and perception skills. Examples of Tangible Interfaces for HCI include the metaDESK[18], mediaBlocks[10], Triangles[7], and Illuminating Light [20]. Because many users can manipulate the physical objects in a Tangible Interface simultaneously, Tangible Interfaces already begin to address issues of co-located multi-user interactions.

This paper has introduced Synchronized Distributed Objects as a way to extend the Tangible Interface approach into distributed CSCW. We have described PSyBench, a shared *physical* workspace over distance, as an application of this concept to shared workspace design. We have also considered applications to physical interpersonal space with inTouch, which allows distant users to communicate using their sense of touch.

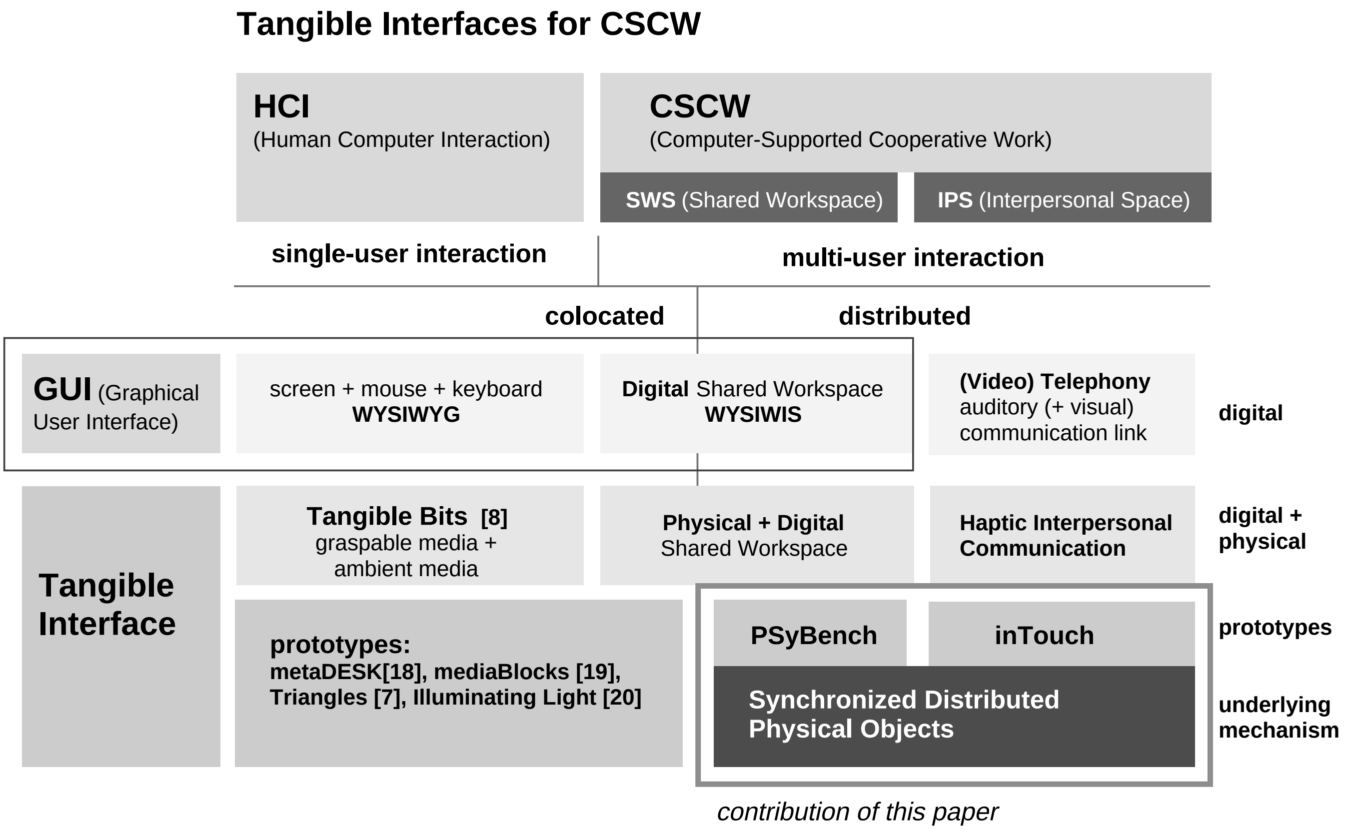


Figure 12. Conceptual framework addressed in this paper.

APPENDIX B: INTOUCH CONTROL ALGORITHM

This appendix describes the control algorithm used for inTouch-1 and inTouch-2 in greater detail.

inTouch-1: Standalone Prototype

As mentioned previously, the control algorithm connects corresponding rollers with a simulated, highly damped, stiff rotary spring. The equations to control a single pair of synchronized rollers is shown below:

$$\tau_0 = -K(\theta_0 - \theta_1) - B(\dot{\theta}_0 - \dot{\theta}_1)$$
$$\tau_1 = -K(\theta_1 - \theta_0) - B(\dot{\theta}_1 - \dot{\theta}_0)$$

$\theta_{0/1}$ = angular positions of the two “connected” rollers

$\tau_{0/1}$ = torque to exert on the corresponding roller

K = spring constant

B = damping constant

Since the system architecture uses only optical position encoders for sensing, angular velocity is interpolated from the ten most recent position readings. Rollover of theta is corrected for so that the rollers behave as expected. It should be noted that the algorithm is symmetrical, giving no roller any advantage over its partner roller.

Optimization

To simulate the direct mechanical connection of inTouch-0 as closely as possible, we would ideally like to set the spring constant (K) extremely high. This constant, however, is limited by the discrete nature of the control algorithm (discrete position encoding, force output, and update interval). Too high of a spring constant for the given parameters will result in unwanted vibration. The maximum torque value is also limited by the strength of the motors.

With the control algorithm running at an update rate of 1 kHz, a spring constant equivalent to ~23mNm/rad gave excellent response and no unwanted vibrations. The maximum output torque of 140nNm for the Maxon motors was also high enough to give an excellent feeling of connection. It should be noted that finite K and maximum torque allow connected rollers to be forced apart from their consistent state; however, doing so merely results in a high force attempting to restore both rollers to that consistent state without causing any harm to the mechanical or control systems. The damping constant, B, was set so that the system appeared to be near critically damped.

Synchronizing More Than Two Objects

A slight remanipulation of the control equations makes clear how to extend the algorithm to synchronize more than two objects:

$$\tau_0 = -2K(\theta_0 - \frac{\theta_0 + \theta_1}{2}) - 2B(\dot{\theta}_0 - \frac{\dot{\theta}_0 + \dot{\theta}_1}{2})$$
$$\tau_1 = -2K(\theta_1 - \frac{\theta_1 + \theta_0}{2}) - 2B(\dot{\theta}_1 - \frac{\dot{\theta}_1 + \dot{\theta}_0}{2})$$

The equations can now also be seen as applying a restoring force on each roller proportional to its offset from the average position of the two rollers. We could clearly now extend this to three rollers for example, but applying a restoring force on each of the three "connected" rollers proportional to its offset from the average position of the three.

inTouch-2: Networked Prototype

As stated earlier, the basic control algorithm for the networked design, inTouch-2, is the same as the algorithm for inTouch-1. Each computer simply calculates the forces to impart to its rollers given the state of each local roller and the most recently received position and velocity of the corresponding remote roller:

Computer 0 runs:

$$\tau_0[t] = -K(\theta_0[t] - \theta_1[t-D]) - B(\dot{\theta}_0[t] - \dot{\theta}_1[t-D])$$

Computer 1 runs:

$$\tau_1[t] = -K(\theta_1[t] - \theta_0[t-D]) - B(\dot{\theta}_1[t] - \dot{\theta}_0[t-D])$$

t = time

D = communication latency (delay)

UDP was chosen as the protocol for communication between distributed objects because it is faster than Transmission Control Protocol (TCP) and the system does not require the reliability of TCP. Absolute position is passed between computers so that a dropped value results in no real loss of data; current values can be assumed to be valid until new values are received. Values are passed between computers along with a count so that values received out of order (i.e. values received that have a lower count than the highest count received so far) are ignored.

Minimizing the Effect of Delay

We have so far distributed inTouch-2 over the local area network in our building (average one-way UDP delay ~2ms). With this small delay, the basic control algorithm described in the previous section works extremely well. Compared to the standalone prototype, inTouch-1, the one difference in performance is that there appears to be more friction on the rollers in the distributed setup. With inTouch-1, the rollers could spin freely (a moderate push would keep a roller spinning for several seconds), while with inTouch-2 the rollers were much harder to spin. The reason for this is that the communication delay causes the local control algorithm to see the remote roller a few steps behind where it really is. So if a user spins a local roller, even if the remote roller is trying to keep up, the local setup sees it as dragging behind, resulting in a resistive force. Our solution to this problem was simply to add prediction into the algorithm so that the local setup is always estimating the true position of the remote roller given the old information. In simulating longer delays, this solution worked well up to a delay of around 12ms (approximate average on-way UDP trip from MIT to University of Pennsylvania), again allowing rollers to spin freely.

Once the delay exceeds around 12ms, however, we begin to see unwanted oscillations in the system due to feedback. Analysis of collected data has shown that the system hits a "resonant" frequency dependent on the delay and other control parameters. A 40ms delay, with the parameters described earlier, for example, results in an unwanted oscillation at around 15Hz. Accurate prediction could alleviate this problem as well, but at these delays, noise in the system compromises the ability to predict accurately and attempting to do so also results in instability.

After recognizing that users rarely try to oscillate the rollers at higher than 5Hz, we decided to put a simple low-pass filter on the position information before it is passed over the network. This solution coupled with a decrease in the spring constant K to 1/3 its previous value stabilized the system up to a delay of 40ms (approximate average on-way UDP trip from MIT to Stanford University). We then added a small amount of prediction back in to alleviate unwanted drag.

Although we had to make the compromise of decreased responsiveness in the system by using a smaller spring constant, K, and a low-pass filter on positions, we were able to achieve very reasonable performance for a delay approximately representing communication across the United States. Since this was achieved with very crude prediction and low-pass filtering, it is likely that further system analysis and tailoring of the control algorithm could increase the allowable delay significantly.