Computer Supported Cooperative Work (CSCW) (2015) 24:109–119
DOI 10.1007/s10606-015-9219-4

Tracking Deictic Gestures over Large Interactive Surfaces

Ali Alavi & Andreas Kunz
Innovation Center Virtual Reality, ETH Zurich, Zurich, Switzerland (E-mail: alavis@ethz.ch; E-mail: kunz@iwf.mavt.ethz.ch)

Abstract. In a collaborative environment, non-verbal communication elements carry important contents. These contents are partially or completely lost in remote collaboration. This paper presents a system to address this issue by tracking pointing gestures, the main non-verbal communication element prevalent in such meetings. The setup employs a touchscreen tabletop computer system for representing the visual content of the meeting, together with three motion trackers for tracking the pointing gestures.

Keywords: Computer supported collaborative work, Non-verbal communication, Remote collaboration, Tabletop computing, Gesture detection

1. Introduction

Innovative ideas typically originate from a collaborative brainstorming within a collocated team as described by Sutton and Hargadon (1996). As described by Gaver et al. (1993), in such a collaboration people focus on the shared artifacts e.g., on the table, and the collaborators use pointing gestures (deictic gestures) to refer to certain artifacts in the common workspace. The importance of non-verbal communication was already researched earlier by Ishii and Kobayashi (1992), Kirk and Stanton Fraser (2006), Kirk et al. (2007), Louwerse and Bangerter (2005), and Tang (1991). It was shown by Kunz et al. (2014) that the interaction between humans and the digital media happens on the "task space", which can be a tabletop computer, as well as above it in the so-called "communication space". The importance of gestures was underlined in a study by Bly (1988). She used two video links to transfer the content of task and communication space, which did not allow editing the content remotely. However, by providing visual contact between two remote stations instead of audio only, she figured out that "gestures constituted a significant portion of the drawing actions that took place". This statement is in line with finding by Gross (2013), who pointed out the importance of awareness in CSCW.

Pointing gestures for example are in the communication space, but they refer to an artifact on the task space. If this context between the two spaces gets lost, the whole gesture will become meaningless. However, today's electronic brainstorming systems are not able to transfer these deictic gestures. Thus, it is important that these pointing gestures are captured, aligned to the artifacts, and correctly transferred to the remote side. However, pointing gestures typically occur in the *communication space* as stated by Kunz et al. (2014) and cannot be tracked by any sensors in the table, since they do not touch the surface. These gestures cannot be replaced by touch interaction neither, since touch interactions are used to as a form of input the underlying software (selecting, moving and so on).

2. Related work

Aligning task space and communication space was already tried earlier. Krueger (1983) gave an early example of such systems when describing a shared workspace. However, since it was not possible to interact with the shared artifacts, it was more a shared view space, i.e., the focus was more on information distribution than on information generation. This problem was addressed later again by Tang and Minneman (1991a' b). They present VideoDraw, a system that allows sharing a common workspace. In a symmetric setup, a camera faces downwards onto the screen, while the captured video image is transferred to the remote side. The partners used whiteboard markers to draw directly on the screen, and thus the camera captured the artifacts together with the drawing gestures. However, moving or deleting objects could only be done locally, and thus a full collaboration was not possible.

Instead of monitors, VideoWhiteboard by Tang and Minneman (1991a, 1991a' b) employed rear-projection and rear-cameras. While the cameras could see clearly the artifacts generated by the regular whiteboard markers, any gesturing of the user in front of the whiteboard could only be detected (and transferred) as a shadow. However, a full control over all generated artifacts was still not possible. Moreover, shadows were often not very clear depending on the distance of the user to the screen.

In order to allow for a workspace that could be edited by both partners, Bly and Minnemann (1990) developed Commune. The system consisted of interconnected horizontal digitizers on top of a horizontally mounted CRT monitor. Although the system offers a common task space, the communication space was supported by audio only, since a video capturing of the remote partner was missing.

ClearBoard by Ishii and Kobayashi (1992) was one of the first systems that brought together task space and communication space. The system allowed partners to see each other, while working on an interactive surface by employing a semi-transparent mirror as an optical combiner of rear-projection and camera-capturing. However, the "content-on-video" metaphor was an unusual way to represent the generated artifact together with the video image of the remote side.

When researching the importance of hand gestures, Kirk and Stanton Fraser (2006) and Kirk et al. (2007) used an asymmetric setup in a worker-helper scenario. Drawing and gesturing was captured by a camera and displayed on the remote side in different geometric alignments. The system was not capable of a full collaborative editing of a shared common workspace.

The idea of “content-on-video” was also realized by Stotts et al. (2003). The system uses a camera to capture face and gestures of the remote partner. The hand gestures can further be used to control the computer’s mouse pointer. However, the system was not designed for an on-screen interaction, since all gestures had to be done in free space.

With “Digital Desk” by Wellner (1993) and “Double Digital Desk” by Wellner and Freeman (1993), they introduced a system that uses a front-projection onto a table as well as a camera mounted above the table. The system was capable of capturing paper-based artifacts and gestures, and to combine them with the digital information of the remote side. Interaction with the system was possible by using mouse or stylus on a tablet, but also fingerpointing was possible through image processing of the acquired camera image. However, the remote station was obviously not able to modify the physical content of the common workspace.

The idea of the Digital Desk was further developed by Kuzuoka et al. (1999) in the Agora system. The system allows for mutual eye contact by adding two vertical screens with an integrated camera, but still does not allow full control over the common workspace.

In order to overcome the problem of limited control over the shared workspace, VideoArms, (Tang et al. 2004; Tang et al. 2006) employed a digital whiteboard, which allowed a shared editing of all generated artifacts. In addition, live-video embodiments representing pointing gestures could be overlaid on the common workspace. Thus, deictic gestures on shared artifacts could be correctly represented. However, due to the real-time constraints the resolution of the video overlay was limited.

The live-video embodiment was improved in the CollaBoard system by Kunz et al. (2010) and Nescher and Kunz (2011), which benefits from the fact that an LCD emits linearly polarized light. Placing an additional linear polarization filter that is rotated by 90° in front of the camera will blind it for the content on the screen, while the user is still visible. This allows separating a person in front of a highly dynamic background on the LCD.

3. System setup

Many of the systems mentioned in the above that are capable of tracking gestures in the communication space, detect and interpret the gesture by augmenting a two-dimensional image of the gesture into the task space. Also our setup follows the recommendation from Gutwin and Greenberg (2002) “to pick up what their colleagues are doing (or not doing) and to adjust their own individual activities accordingly”. This means that the system supports the users’ needs of displaying and monitoring activities, as stated by Schmidt (2002). More specifically, our system allows capturing and transferring deictic gestures that are related to visual content. However, this will lead to ambiguity whenever multiple artifacts are in the pointing direction. To detect such gestures in the 3D space more reliably, it is not sufficient to just orthogonally map the position of the fingertips onto the interactive surface in order to achieve x- and y-coordinates, but also the height (z-coordinate) of the fingertip is of importance. If in addition a second measurement point of the pointing gesture could be achieved, it would be possible to represent the pointing gesture as a vector, which has a well-defined intersection point with an artifact on the task space. Thus, we need to capture 3D positions and orientations of the gestures. While this can be achieved by depth sensing cameras such as Kinect, such cameras should be set up above the tabletop in order to view the pointing gestures performed by all users around the table. This complicates the setup of the system. Moreover, many such cameras work with infrared light, which might interfere with tabletops using FTIR or other infrared imaging technologies. Available solutions for this problem reduce the depth resolution of the depth sensing camera to a level which makes the camera useless for gesture tracking as shown by Kunz et al. (2014).

Due to abovementioned limitations, we decided to use one depth-sensing camera per user. In this way, we can set up our system without facing those problems: our system setup consists of a tabletop touchscreen, namely Microsoft PixelSense. Three Leap Motion sensors are placed on the border of the table, enabling tracking hand gestures above the surface of three persons standing at the corresponding sides of the table (Figure 1). This setup is easier to realize than a camera-from-the-top solution. Moreover, the inclination of the LEAP Motion (LEAP Motion 2014) sensors was chosen in such a way that they do not see the Pixelsense's surface, hence eliminating any interference. Each sensor is oriented in such a way that one edge of its viewing cone is parallel to PixelSense's surface, allowing for the best detection of pointing gestures. Also, the large distance between the two sensors facing each other prevents any interference between them.

The gestures are displayed at the remote side as a highlighter. The remote side uses a regular computer and a mouse; a videoconferencing is not required, but only an audio connection. Since the remote side is not expected to perform deictic gestures, this asymmetric setup does not influence the results of the user study.

Since Leap Motion sensors need dedicated computer systems, we have to use individual computers for each sensor and send the data over a network. We used a publisher-subscriber pattern, where the computers connected to the sensors act as publishers, and PixelSense acts as subscriber (Figure 2). We implemented this model using umondo by Aitenbichler et al. (2007), a library for rapid development of publish-subscribe distributed software. All the mentioned software is developed for Microsoft Windows 7 using Microsoft C#.

3.1. Calibration

In order to calculate the target of the pointing gestures on the screen, our tracking algorithm first needs to know the relative position of each Leap Motion with regard to PixelSense. This is done during the calibration phase. For calibrating the system,

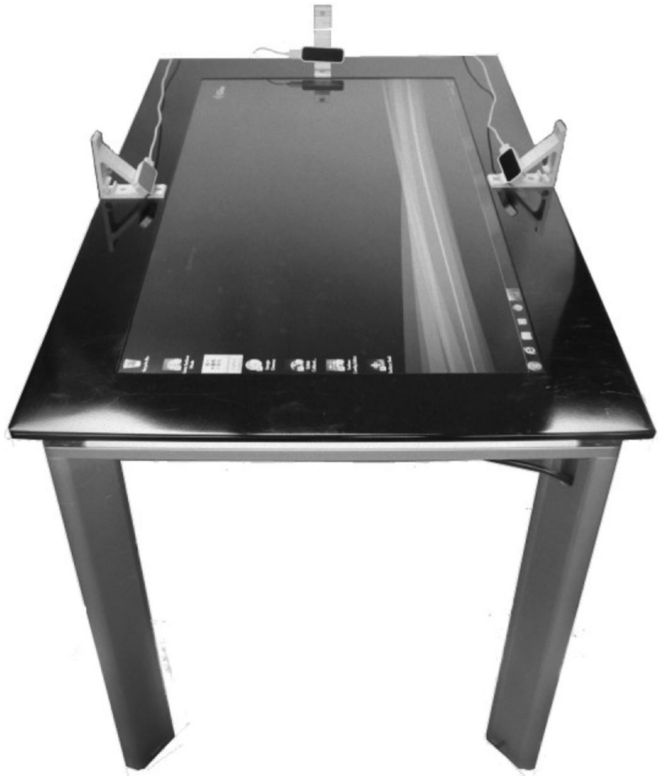

Figure 1. Test setup of the overall system. Note that the sensors are still on a large stand to test various inclination angles.

the user has to touch the screen, which is captured by both the PixelSense and the Leap Motion in front the user. Since PixelSense and Leap Motion have their own coordinate system, these systems need to be transformed to a common one (see Figure 3) in order to compare the individual measures of both sensors. Moreover,

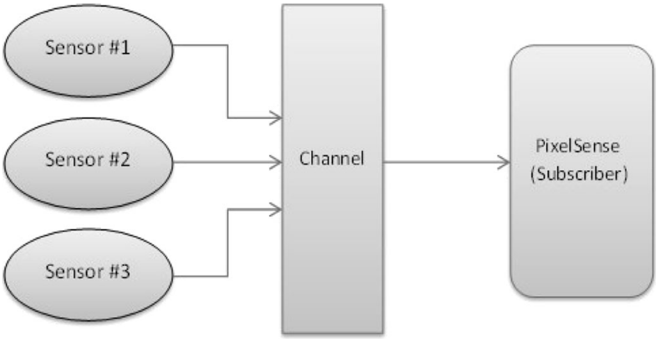

Figure 2. Publisher-Subscriber pattern used for using multiple leaps

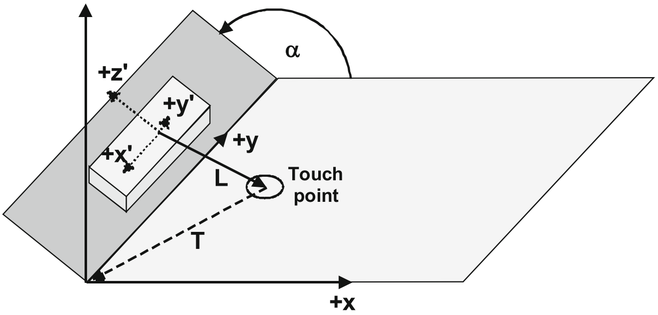

Figure 3. Schematics of calibration phase for a single Leap Motion sensor. Observe that vector L belongs to Leap Motion's coordinate system, while vector T belongs to the PixelSense coordinate system.

Leap Motion uses a metric coordinate system, while PixelSense's unit is in pixels. After performing these transformations, the system compares the calculated touch point with the data captured from the touch screen, in order to find the constant shifts and slope of the calibration. The calibration process has to be done for all LEAP sensors that were placed on the table (Figure 1).

After the calibration, the inclination and rotation angles of the pointing finger, as well as its tip position, are transformed to the coordinate system of PixelSense.

$$\begin{aligned} X &= X' - Z' \cos(\alpha) - Y' \sin(\alpha) \\ Y &= Y' + X' \\ Z &= -Z' \sin(\alpha) + Y' \cos(a) \end{aligned}$$

Then, the intersection of this vector and PixelSense plane (z=0) defines the pointing gesture's target on the screen.

When performing a pointing gesture onto a certain target, the user is supported in his pointing action by a visual highlighter that will appear at the intersection point mentioned in the above. This highlighter helps the user to precisely select an object on the screen.

4. Experiment

The goal of the experiment was to show that a net-based collaboration with pointing gestures outperforms a voice-based communication. Thus, the task has to be designed in such a way that it cannot be easily solved verbally, but requires non-verbal communication means such as pointing gestures. However, instead of augmenting the full image of the pointing gesture onto the remote side's screen, the target position of the pointing gestures is overlaid. While the detection of in-air gestures is expected to be superior to a verbal description of the position, it won't make a difference to the remote partner who simply sees the highlighter together with an audio command.

4.1. Design

To evaluate how transmission of pointing gestures affects the performance of a collaborative work, we designed a simple experiment in which users have to participate in a remote collaboration task. The users can use video and voice conferencing using Skype, although a video connection is not required. The task involves coloring of a figure. One partner has a colored figure, while the other one has a similar, uncolored figure. The first partner should describe the coloring of the figure to the remote partner, so that he or she can paint the figure correctly (Figure 4). Each white field can be colored separately, requiring either a precise pointing gesture or an exact (but probably longer) verbal description of the element that should be colored next. Partners had to make sure that all fields will be colored. However, the instructing partner had no visual feedback whether the remote person colored all fields and thus had to ask what is missing. The task was completed when all fields are colored and then the completion time was measured.

We designed the user study in such a way that each participant had to color the fields of the object twice. In the first test, he was instructed by pointing gestures, while in the second test he only perceived a verbal instruction. For example, the partner might say: *the vase is blue*, *or the leaf on the left side of the vase is green*. In order to avoid any biasing of the results, we changed the order of the tests as well as the color palette by inverting the colors of the first image. This assures that the tasks have the same level of difficulty, (i.e., number of colors). Prior to each test, there was also a short instruction on how to select colors from the palette and how to apply them to the object.

4.2. Hypotheses

Prior to the experiments, the following hypotheses were stated:

- H1: The completion time is mainly defined by handling the painting program, thus no clear difference between pointing gestures and verbal explanations are visible (Null hypothesis)

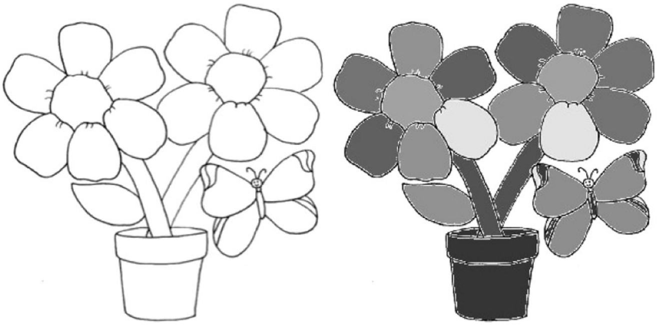

Figure 4. Image used for the experiment. One partner sees the colored image (*right*) and instructs the other partner who only sees the uncolored image.

- H2: The remote user will get irritated by the highlighter due to an unstable position and thus will perform the wrong action, i.e., he will colorize the wrong field of the object. This will result in an additional clarification effort and thus in a longer completion time than for verbal explanations.
- H3: The pointing instruction outperforms the verbal instruction, since the object is already too complex to vastly describe the corresponding field by audio only.

4.3. Participants

Nine participants took part in this study, including eight male and one female. None of them had any color blindness. Each of them participated in separate experiments. None of the participants was able to communicate in his or her mother's tongue, but used English as common communication platform. All participants had at least a communicative set of commands in English.

5. Results

We performed the experiment using two different setups: one without the tracking system, in which the users can only communicate using the video conferencing feature, and one using the proposed setup, during which the remote users can see the target of the partner's pointing gesture. We measured the completion time of the task using these two different setups (Figure 5).

The user study showed that our designed system is capable of supporting deictic gestures, since it outperforms the communication between remote partners by a significantly shorter completion time. As shown in Figure 5, the mean completion time for the task with pointing is by 32 % shorter than the task that used verbal descriptions only. Moreover, the variance for the pointing task is significantly smaller

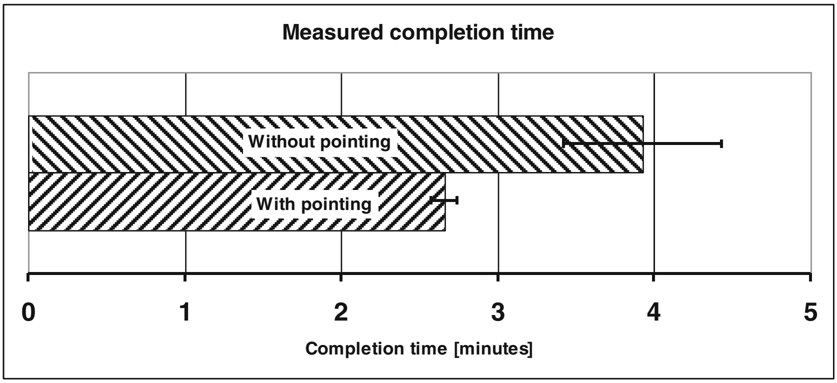

Figure 5. Completion time for the two different setups.

(0.16) than for the task without pointing (1.03). This can be explained by the fact that pointing gestures are unequivocal compared to a verbal description. Moreover, the large variance in the verbal-only task is mainly due to the fact that it was difficult for users to describe the geometry and position by words, which in some cases also resulted in certain misunderstandings that had to be clarified.

The hypothesis H1 turned out to be non-valid, since there was a significant difference (32 %) in the completion times between the task with pointing and the one without pointing. Consequently, the effect of handling of the paint program can be neglected.

Also hypothesis H2 did not hold true, since in all cases the completion time for the task with pointing gestures was shorter. This means that irritations of the highlighter did not occur at all or only had a minor influence on the completion time.

Finally, only hypothesis H3 turned out to be true, since pointing gestures are suitable to describe positions on the screen much faster than it could be done by a verbal explanation. Thanks to the highlighter that is controlled by the pointing gesture, it gives unique information about the object of interest. This is not only because of the fact that it can be easily understood by the remote partner, but also because the highlighter gives the possibility to intuitively correct the pointing gestures so that it gives precise information. Thus, the system could also be used for coordinative tasks like e.g., in air traffic control as described Berndtsson & Normark (1999).

6. Conclusion and future work

Within this paper, we presented a system for tracking pointing gesture in collaborative tabletop scenarios using Leap Motion sensors (with the corresponding PCs) and Microsoft PixelSense tabletop computing system. Such setup enables easy integration of gesture detection into tabletop collaboration. We evaluated our setup by performing a user study, involving an asymmetric remote collaborative task. The tasks involved coloring of an uncolored image by a desktop computer user, while the instructions are gives over the network by a tabletop computer user. The results show significant performance improvement when the gesture detection system is used. On average, users managed to perform the task 32 % faster when their pointing gestures were tracked using the proposed setup, comparing to an audio only remote collaboration.

Future work will focus on integrating other gestures into the system. In this step, not only capturing these gestures, but also their remote representation can be interesting research questions. Moreover, we will improve the noise filtering of the tracking signals. Currently, we use the raw data of the sensors, which are noisy and sometimes completely loose the pointing finger for some milliseconds. Since pointing gestures are not time-critical, we will apply an exponential or double-exponential smoothing to the signals, which hopefully will result in a steady tracking signal. This will also reduce the jitter of the highlighter and eventually the user irritation. After the implementation of this signal filtering, we will also integrate other gestures than pointing. This will make the system also suitable for many other applications, such as brainstorming.

Acknowledgments

This work was done within the project CR21I2L_138601 of the Swiss National Science Foundation.

References

Aitenbichler, Erwin, Jussi Kangasharju, and Max Mühlhäuser (2007). MundoCore: A Light-weight Infrastructure for Pervasive Computing. *Journal of Pervasive and Mobile Computing*, vol. 3, no. 4, pp. 332–361.

Berndtsson, Johan, and Maria Normark (1999). The coordinative functions of flight strips: Air Traffic Control work revisited. *GROUP'99: International Conference on Supporting Group Work, Phoenix, Arizona, 14–17 November 1999,* New York: ACM Press, pp. 101–110.

Bly, Sara (1988). A Use of Drawing Surfaces in Different Collaborative Settings. *CSCW'88 Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, New York: ACM Press, pp. 250–258.

Bly, Sara A., and Scott L. Minnemann (1990). Commune: A Shared Drawing Surface. *COCS'90 Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems*, New York: ACM Press, pp. 184–192.

Gaver, William W., Abigail Sellen, Christian Heath, and Paul Luff (1993). One is Not Enough: Multiple Views in a Media Space. *CHI'93 Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 335–341.

Gross, Tom (2013). Supporting effortless coordination: 25 years of awareness research. *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing and Work Practices*, vol. 22, nos. 4–6, 1. August 2013, pp. 425–474.

Gutwin, Carl, and Saul Greenberg (2002). A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, vol. 11, nos. 3–4, pp. 411–446.

Ishii, Hiroshi, and Minoru Kobayashi (1992). ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. *CHI'92 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 525–532.

Kirk, David., Tom Rodden, and Danea Stanton Fraser (2007). Turn It This Way: Grounding Collaborative Action with Remote Gestures. *CHI'07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 1039–1048.

Kirk, David, and Danea Stanton Fraser (2006). Comparing Remote Gesture Technologies for Supporting Collaborative Physical Tasks. *CHI'06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: ACM Press, pp.1191–1200.

Krueger, Myron (1983). *Artificial Reality*, Addison-Wesley Professional

Kunz, Andreas, Thomas Nescher, and Martin Küchler (2010). CollaBoard: A Novel Interactive Whiteboard for Remote Collaboration with People on Content. *CW 2010 Proceeding of the International Conference on Cyberworlds*, 20. – 22. October 2010, IEEE, pp. 430–437.

Kunz, Andreas, Ali Alavi, and Philipp Sinn (2014). Integrating Pointing Gesture Detection for Enhancing Brainstorming Meetings Using Kinect and PixelSense. *8th International Conference on Digital Enterprise Technology*, Stuttgart, Germany, 25. – 28. March 2014, pp. 1–8.

Kuzuoka, Hideaki, Jun Yamashita, Keiichi Yamazaki, and Akiko Yamazaki (1999). Agora: A Remote Collaboration System that Enables Mutual Monitoring. *CHI EA'99 CHI'99 Extended Abstracts on Human Factors in Computing Systems*, New York: ACM Press, 15. – 20. May 1999, pp. 190–191.

LEAP Motion. https://www.leapmotion.com. Accessed 10. March 2014.

Louwerse, Max, and Adrian Bangerter (2005). Focusing Attention with Deictic Gestures and Linguistic Expressions. *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, 23 pp. 1331–1336.

Nescher, Thomas, and Andreas Kunz (2011). An Interactive Whiteboard for Immersive Telecollaboration. *The Visual Computer: International Journal of Computer Graphics - Special Issue on CYBERWORLDS 2010*, vol. 27, no. 4, April 2011, New York: Springer, pp. 311–320.

Schmidt, Kjeld (2002). The problem with "awareness": Introductory remarks on "Awareness in CSCW". *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing*, vol. 11, nos. 3–4, pp. 285–298.

Stotts, David, Jason McC. Smith, and D. Jen (2003). The Vis-a-Vid Transparent Video Facetop. *Proceedings of the UIST 2003*, pp. 57–58.

Sutton, Robert, and Andrew Hargadon (1996). Brainstorming Groups in Context: Effectiveness in a Product Design Film. *Administrative Science Quarterly*, vol. 41, no. 4, pp. 685–718.

Tang, John C. (1991). Findings from Observational Studies of Collaborative Work. *International Journal Man–machine Studies*, vol. 34, no. 2, pp. 143–160.

Tang, John C., and Scott L. Minneman (1991). VideoDraw: A Video Interface for Collaborative Drawing. *ACM Transactions on Information Systems (TOIS) - Special issue on computer—human interaction,* vol. 9, no. 2, New York: ACM Press, April 1991, pp. 170–184.

Tang, John C., and Scott L. Minneman (1991). VideoWhiteboard: Video Shadows to Support Remote Collaboration. *CHI'91 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 315–322.

Tang, Anthony, Carman Neustaedter, and Saul Greenberg (2004). VideoArms: Supporting Remote Embodiment in Groupware. *Video Proceedings of the ACM Conference on Computer Supported Cooperative Work - ACM CSCW'04*, New York: ACM Press.

Tang, Anthony, Carman Neustaedter, and Saul Green (2006). VideoArms: Embodiments for Mixed Presence Groupware. *Proceedings of HCI 2006*, London: Springer, pp. 85–102.

Wellner, Pierre (1993). Interaction with Paper on the Digital Desk. *Communications of the ACM - Special issue on computer augmented environments: back to the real world*, vol. 36, no. 7, New York: ACM Press, July 1993, pp. 87–96.

Wellner, Pierre, and Stephen Freeman (1993). *The Double Digital Desk: Shared Editing of Paper Documents*. XEROX Euro PARC Technical Report EPC-93-108.