**Proceedings of the Second IASTED International Conference**
**HUMAN-COMPUTER INTERACTION**
**March 14–16, 2007, Chamonix, France**
**ISBN Hardcopy: 978-0-88986-654-6 / CD: 978-0-88986-655-3**

# REMOTE GESTURE VISUALIZATION FOR EFFICIENT DISTANT COLLABORATION USING COLLOCATED SHARED INTERFACES

François Coldefy and Stéphane Louis-dit-Picard
France Télécom R&D
2, Av. Pierre Marzin
22300 Lannion Cedex 9, France
email: {francois.coldefy, stephane.louisditpicard}@orange-ftgroup.com

## ABSTRACT

Remote gesture visualization contributes to the efficiency of distant collaboration tasks because it enables the coordination among participant's actions and talk. This paper presents an efficient collaborative system combining Computer Vision Techniques and Collaborative Tools applied to multi-site connected shared table interfaces (MERL Diamond Touch). This platform provides a fluid gesture visualization of each distant participant on or above the tactile table, whether he/she is moving virtual digital objects or is intending to do so. Our main contribution addresses technical improvements in the domain of the " side by side" metaphor for Computer Supported Cooperative Work (CSCW), in terms of real time computer vision processing, robustness, constraint relaxation on lightning conditions and shared workspace content (images, videos, manipulation of windows). This system may be extended to any kind of interactive displays (projection screens or tactile PC monitors

## KEY WORDS

Computer Supported Cooperative Work (CSCW). Devices and display systems, tools and interaction techniques

## 1. Remote gesture visualization

Remote gesture visualization brings significant improvements in performing distant collaborative tasks because it allows the coordination among participant's action and talk [1]. As Kraut *et al.* [2] state, visual information plays at least two inter-related roles in collaborative tasks: situational awareness and conversational grounding. First, visual information conveys information helping people to maintain up-to-date mental models of complex and dynamic environments. It helps to be aware of the state of task objects and of one another's activities. Situational awareness facilitates to plan what to say or do next and to coordinate speech and actions with distant participants. Second, visual information helps people to communicate about the task to be done by ensuring that their message is properly understood. It enables the expansion of their common ground during the session and facilitates mutual understanding between participants. There is significantly less talk about the talk (or about the task to be done)[1].

In case of distant collaboration between collocated shared interfaces for which several users may interact simultaneously on each distant site, remote gesture visualization is particularly effective to provide a higher level of mutual awareness because it enables a visual identification of the distant interacting users. Furthermore, it helps to preserve common social rules within the interaction. It avoids unwanted simultaneous actions on a same object as one would not take an object right from partner's hands.

In this paper, we present a remote gesture visualization device for distant table interactions. In section 2, we present a state of the art of remote gesture and of user visualization systems for collaborative applications. Section 3 provides a description of the network architecture and of the tabletop we used. Section 4 focuses on computer vision techniques to enhance remote gesture visualization. In Section 5, we present some experimental results. Finally, section 6 draws conclusions and perspectives for future work.

## 2. Related work

Remote gesture and/or user visualization systems for collaborative tasks were developed up to now according to three major metaphors reported by Ishii *et al.* [3] in 1992:

a) the “side by side” metaphor: distant users are “side by side” in front of a same board or table,

b) the “transparent glass window” metaphor in which distant users are on both sides of an augmented glass,

c) the “face to face table” metaphor in which distant users are sitting on both sides of a table and talking face to face and interacting on and over the table.

Whereas the first two metaphors essentially imply video and computer vision techniques, the table metaphor often deals with virtual or mixed reality technology. In the state of the art below, we focus on integrated systems for communication and collaboration and exclude communication systems which do not involve shared workspaces.

### 2.1 The “side by side” metaphor

In 1991, VideoDraw [4] was a first video based attempt to capture participant's gesture and action for a remote shared drawing application. Each participant was drawing directly on the display screen of a video monitor with whiteboard markers. A camera, aimed at the display screen, transmitted to the other site the captured image, *i.e.* the participant's marks and gestures. Polarized filters were used to manage video feedback. However, no digital collaboration was afforded as VideoDraw was based on solely analogical video techniques. The image resolution was poor at this time and the screen size limited.

More recently, Roussel [5] proposed in 2001 a shared telepointer device using chroma-keying to extract the user's hand placed over a blue surface and overlaid on the common view of the shared space. Malik *et al.* [6] presented Visual TouchPad a more sophisticated device based on 3D tracking of both hands on a sensitive blue pad for tactile and 3D interaction. The shared desktop image is then augmented by the insertion of the both hands image. However, the user is not interacting on the visualization surface itself as he or she moves his/her hand on a blue pad.

In 2004, Takao proposes Tele-Graffiti [7], a remote sketching system for distant shared writings on paper sheets. It allows two distant users to contribute to a virtual leaf, fusing the marks each participant has written on his/her actual paper sheet. The system is composed at each site of a camera-projector couple aimed at the white paper leaf on which the user writes or draws. The user's actions and marks are captured by the camera and are transmitted to the distant site where the image is projected on the other user's sheet after a dynamic geometric alignment. As minimal computer vision analysis of the projected and captured images is performed, the system runs at video frame rate (24 Hz) but a video feedback or visual echoing may occur. Thus delicate camera and projector calibration and lightning adjustment are needed. Liao *et al.* [8] propose computer vision techniques to remove this visual echo. Unfortunately, the frame rate then drops drastically below 7 Hz.

A very accomplished work in the domain of the side by side gesture visualization is VideoArms proposed by Tang *et al.* [9]. The authors conceived a collaborative system with effective distant mutual awareness by coupling a sensitive SmartBoard with a video capture of participant's gesture overlaid on the remote desktop image. Users can easily predict, understand and interpret distant users' actions or intents as their arms are visible whenever they want to interact with the tactile surface or show anything on it.

### 2.2 The “transparent glass window” metaphor

A second group of remote gesture visualization systems is based on the transparent glass window metaphor. Distant users share an interaction surface of various size (a video monitor for ClearBoard [3] or a large projection screen for VideoWhiteBoard [10] and a later version of ClearBoard [11]) as if they were on both sides.

In order to enlarge the drastically limited shared drawing surface of VideoDraw, Tang *et al.* proposed VideoWhiteBoard [10], a system using translucent large rear projection screens while a camera mounted behind the screen captured both drawings and people shadows. But VideoWhiteBoard prevented distant user identification as only his or her shadow was transmitted. Thus, it did not provide any help to collaborative tasks when several persons per site were involved.

ClearBoard [3] enabled eye contact between distant participants by using a half mirror polarizing projection screen whose transparency allows rear video projection while user's reflected image is captured by a camera. As for VideoDraw, the user was drawing directly on the display surface which shows user's distant drawings fused with user's face. An other ClearBoard version [11] was a first arrangement in coupling remote gesture visualization and digital collaboration by using a digitizing pen.

In 2003, Stotts *et al* [12] proposed Vis-a-Vid, a video facetop dedicated to pair programming and consisting in an overlay of the distant user video on the image desktop. Hand tracking for gestural interaction was implemented to enable distant interaction on large projection screens. A later version, FaceTop [13], proposes a more integrated platform to MACOS X allowing various granularity of windows and screen sharing including a glass board functionality for shared annotations. The main drawback is the camera position beside the screen for cost and space convenience. It prevents eye contact between distant participants and implies video feedback to allow the user to see exactly what location his gesture refers to.

Wilson [14] proposed TouchLight, a 3D interactive device which uses a DNP HoloScreen, a refractive holographic film which lets rear projection at a specific angle while it remains transparent to visible light and Infra Red. Object tracking on the interaction surface is performed by binocular stereo vision camera behind the screen. The system should be applied to distant user video visualization as it enables eye contact, but interferences between the light coming through the screen and the projected image prevent up to now such an application.

### 2.3 The “face to face table” metaphor

The face to face table metaphor has been essentially developed in Collaborative Virtual Environments frameworks. Spin-3D [15] and cAR/PE! [16] fostered the “meeting room” metaphor in proposing a shared virtual room in which all virtual objects and users are visible. In [15], users are represented by avatars and are virtually sitting around a virtual table on which 3D objects are placed for collaborative interaction. User object selection and manipulation is visualized by letting the avatar pointing at the manipulated object. Whereas Spin-3D does not handle any navigation inside the virtual room, cAR/PE! proposed predefined views allowing users to “move” according to view changes. cAR/PE! represents distant users with videos integrated around the virtual table. Carpeno [17] (for “cAR/PE!” and “Coeno”) extends the cAR/PE! system to allow remote collaboration using a tabletop: cAR/PE! handles the virtual teleconferencing and the interaction around a shared virtual table, whereas Coeno handles the interaction on both the local shared tabletop and local private spaces (TabletPC, PDA, etc.).

Recently, ViCAT [18] defines a face-to-face visualization and interaction system extending the notion of a physical table to the CSCW. Several users are able to collaborate on a virtual multi-user desktop, displayed on large horizontal screen, and to communicate using videos of remote users displayed on a large vertical screen. Users' interaction with object is symbolized by telepointers.

### 2.4 The side by side metaphor ... around the table

We focus on distant table interaction. TableTop as an input/output device is an exciting and emerging cross disciplinary domain of computer systems involving projector based display systems, augmented reality, user interface technologies, multi-modal and multi-user interactions, CSCW, and information visualisation (see [19] for the first TableTop conference). For distant interaction, we choose the “Side by Side” metaphor partly because Kirk *et al.* [1] showed that this scheme significantly improves remote collaboration. Furthermore, the side by side metaphor appears to be more suitable for table interaction as no direction is favoured anymore. Documents may thus be user oriented as in Diamond Spin [20]. The “transparent glass windows metaphor” adapted by Roussel [21] to horizontal surfaces could be hardly fitted to our device. Finally, most of the experiments based on the “face to face table” metaphor still lacks of the required fluidity of interaction due to the available Augmented Reality system [2].

This paper is a continuation of VideoArms [9]. Our main contribution addresses three technical limitations of VideoArms in terms of Computer Vision: 1) the proposed method can detect any object on or over the table; no learning stage or a priori knowledge is needed to detect hands or tools; 2) no restriction on the projected images is imposed (videoArms needs dark tones images); 3) the proposed method is robust to external lightning changes (variation in the daylight or in the artificial lightning); 4) calibration is automatically processed.

## 3. System description

Our system consists, for each site, of a ceiling projector beaming on the Merl Diamond Touch table [22] and a ceiling camera capturing the projected images. A collaborative architecture provides distant consistency of the table shared work space and remote gesture visualization (see Figure 1). More precisely, it comprises:

- a collaborative application server managing the shared desktop of the connected tables,
- at each site a segmentation process analyzing the camera captured image and producing arms image masks whenever the users are interacting on or above the table,
- an image server which receives from each site the image masks and dispatches them to the remote sites.

In addition, an audio server not described on the figure, connects distant sites. The speech communication channel is used to maintain coordination between the participants.

Although the designed architecture is compatible for several connected tables, we use for experimental set up only two tables.

### 3.1 Collaborative Application Server

On each site, Diamond Touch allows up to four users to interact simultaneously. Interaction with window containers (rotation, translation, resize, zoom, etc.) is modelled according to the ”sheet of paper” metaphor. The collaborative application server is in charge of managing the collaboration between the connected sites. It provides the replication of events occurring on window containers in one site to the remote sites in order to keep the consistency of the shared desktop.

### 3.2 Segmentation process

The segmentation process detects any object above the table by comparison between the camera and the shared desktop images up to a geometric and color warping. In output, it produces an image mask of the detected object seen by the camera (see Section 4 for details).

### 3.3 Image Server

The image server receives independently from each site the local participants' gestures masks. In return, it dispatches it to the other sites for overlay on the local current shared workspace views. The overlay is performed using alphablending in order to allow local participants to see the window containers under the distant users' arms.

### 3.4 Implementation

We use an adhoc system based on Java and JOGL (Java Binding for OpenGL) to implement the interactive window containers displayed on the tabletop as shared workspace. Java 2D (based on the Swing API) manages the rendering of window containers in off-screen buffers that we use as OpenGL textures to compose the desktop displayed on the tabletop. As shown in figure 2, the desktop compositor lays out textured quads (each quad is a window container representing a Java 2D application). Users interact with quads (rotation, translation, scaling) or with their contents (Java 2D applications). Off-screen buffers and their associated OpenGL textures are dynamically updated.

Each shared desktop process is connected to the collaborative application server using a TCP/IP network link.

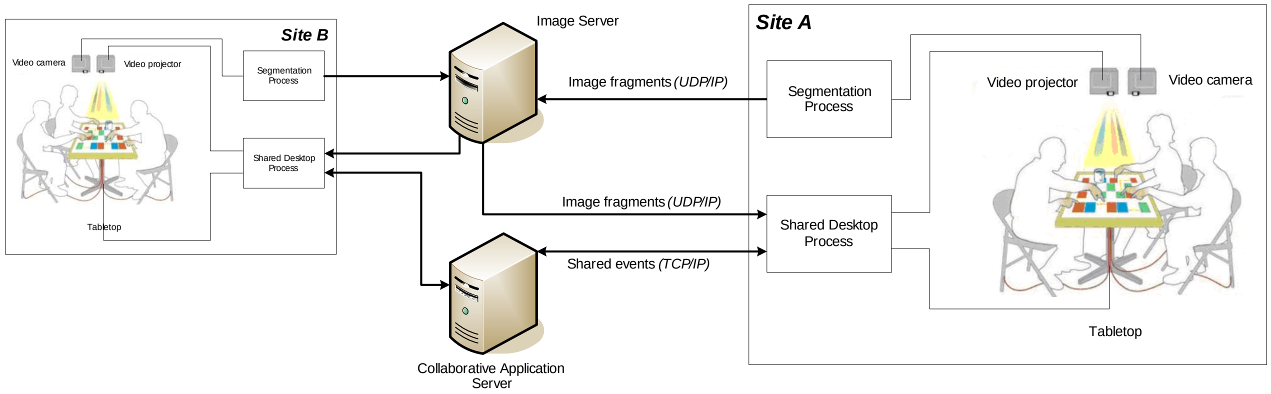

Figure 1. Architecture of the system for two sites (which can be extended to more than two sites)

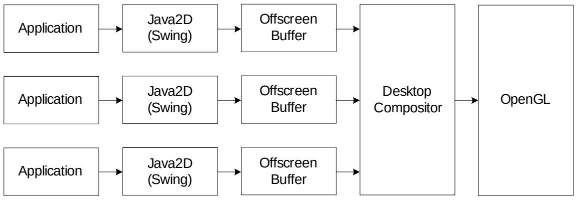

Figure 2. The desktop composition using off-screen buffers of Java 2D/Swing applications and OpenGL

All events (including click, dragging operations, etc.) performed by local participants on window containers are sent to the server which dispatches them to all connected sites. Events received from the network are injected in the event pump of the shared desktop processes to be handled as other local events.

In the segmentation process, each segmented image is compressed using an RLE (Run Length Encoding) encoding technique. As mask images contain very few information, such a coding algorithm is very efficient in time processing and compression. The segmentation process is connected to the image server using a UDP/IP based network link. Each segmented image is split into several fragments wrapped into UDP/IP packets which are sent to remote shared desktop processes through the image server.

Each shared desktop process receives the image fragments and rebuilds the image in order to integrate it as a semi-transparent 2D layer displayed in front of the shared desktop. The segmented image is overlaid on the shared desktop using OpenGL texturing. As UDP/IP is not fully reliable, few image fragments could be missing and so the segmented image could be partially rebuilt. However, as segmented image is considered as a stream, corrupted rebuilt images will be replaced later by a proper one.

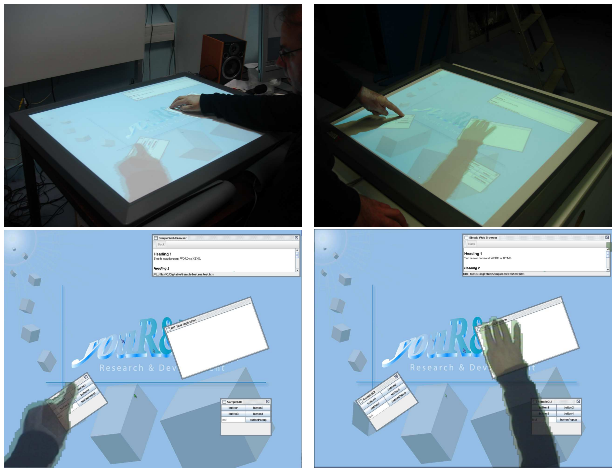

Figure 3. Remote gesture visualisation: view of both distant tables in action (upper line) and of both overlaid desktops (lower line); the left bottom image shows the overlay of the detected hand of site 1 (upper right image); the right bottom image shows the overlay of the detected arms of the site 2 (upper left image)

## 4. Computer vision

The method consists of comparing the captured image in regard with the known projected desktop in order to detect anything (hands, arms, or any object) between the projection surface (here the Diamond Touch) and the camera. Hands and arms are identified as the locus of the image (pixels) where the camera image and the projected desktop do not correspond up to a geometric and color model transformation.

### 4.1 Geometric and color models

We consider that the image captured by the camera corresponds to the known projected desktop onto the table up to an homographic transformation $H$ from $\Re^2$ to $\Re^2$ and a 3-color transfer function $F(c,.)$ from [0,255] to [0,255] where $c$ represents a color channel ($R, G, B$ or $Y, U, V$ indifferently).Let us define $D$ the original desktop image and $R$ the image captured by the camera of the projection of $D$ onto the table. When no object is present above the table, the geometric and color warping between $D$ and $R$ can be expressed at each pixel $s$ of the desktop image grid $S$ and for each color channel $c$ as:

$$R(c, Hs) = F(c, D(c, s)) + \epsilon(c, s), \tag{1}$$

where $\epsilon(c, s)$ is a spatially independent Gaussian noise of mean 0 and unknown standard deviation $\sigma$. Each unknown transfer function $F(c,.)$ is supposed to belongs to $G(K)$, the vector space engendered by the spline functions of order 3 $(g_k)_{k=1..p}$ defined on [0,255] and associated to the knots ensemble $K = \{0, n-1, 2n-1, ..., 255\}$ where $p$ and $n$ are linked by $p = 256/n + 2$. In practice, we use $n = 8$ knots, which seems sufficient to model the colour transfer function. The color transfer function is then expressed as $F(c,.) = \sum_{k=1}^{p} \alpha_{c,k} g_k(.)$ where $\alpha_{c,k}$, are the cp unknown parameters.

### 4.2 Parameters estimation

The homography $H$ (eight unknown parameters) and the $\alpha_{c,k}$ are estimated according to a M-estimator criterion [23]. Let $\rho(.)$ be the Tukey function, and $\theta = (H, \alpha_{c,k})$ be the unknown parameters. We aim at minimizing the cost function $U$ with respect to $\theta$:

$$U(\theta) = \sum_{c=1}^{2} \sum_{s \in S} \rho(\epsilon(c, s)), \tag{2}$$

where $\epsilon(c, s) = R(c, Hs) - \sum_{k=1}^{p} \alpha_{c,k} g_k(D(c, s))$.

### 4.3 Camera Calibration

The homography is first initialized by an automatic procedure: a white desktop image with large green squares at the corners is projected onto the table. The green squares are automatically detected in the camera image by a mean-shift segmentation algorithm and color identification. The four corner positions of the projected desktop in the camera image are then computed. A coarse estimation of the homography $H$ is straight forward computed from the correspondence between desktop corners and their estimated positions in camera image. A second stage refines the above estimation by projecting a contrasted desktop image with marked contours. The homography and the unknown spline parameters of the 3-color transfer functions are computed from an Iterative Reweighed Least Square scheme (IRLS) [23]. However, as the relation 2 is not linear in $H$, the minimization needs an extra loop: at each step, we compute $\theta$ which minimizes a linearized expression of equation 2 by IRLS (see [24] for such a method).

### 4.4 Online object detection

After calibration, $H$ is supposed to be known. The online estimation of the 3-color transfer functions is computed at each time $t$ from one iteration of the IRLS algorithm applied to $D_t$ and $R_t$, respectively the desktop and camera images at time $t$. Note that relation 2 is linear in $\alpha_{c,k}$. This online estimation allows the auto-adaptation of the system to any lightning changes. Any object (a hand, an arm or anything else) placed on the table will mask the projected desktop. It will be detected in the camera image as the set of pixels $s$ for which relation 2 is not valid, i.e. for which the error weight [23] computed in the IRLS iteration is greater than a pre-defined threshold. Some classical post-processing such as morphological filtering is performed to remove small detections.

The local variations of the natural light illumination, especially on larger surfaces such as tables, lead us to estimate the color transfer functions on local patches. We divide the desktop image in four equal patches in which we independently estimate the three color transfer functions for each frame.

Let us notice that equation 2 is expressed in the desktop grid $S$ domain. We sub-sample the desktop image, which is a SXGA (1280x1024) image, to 320x256 in order to reduce time processing.

### 4.5 Implementation

We use multithreading to take advantage of the dual-core processors. We implement one thread for driving the camera capture and the shared desktop acquisition. Another thread, triggered by the previous one waits for desktop and camera images, computes the image masks before transmission to the image server.

## 5. Experiments

We use a Sony EVI-D70 whose pan, tilt and zoom functions facilitates the set up. The desktop image is SXGA (1280x1024) and we choose a medium camera resolution (384x288). Figure 3 shows the remote gesture visualization in action on two distant sites (upper line). The bottom line shows how the local shared desktop looks like at each site, including the overlay of the distant participant arms. No visual echo occurs. However, as camera and desktop captures are not synchronized and are acquired successively, some traces may be visible due to fast gestures or shared workspace changes as fast moving windows or video cuts during film visualization. No constraint on the desktop image is imposed anymore (colors, brightness, etc.). Every kind of object can be detected, if they are thicker than a pen due to the coarse analysis resolution (320x256).

The computer vision algorithm provides 32 image masks per second when running alone. When connected, the arms images mask are refreshed on each site at 17 Hz on a dual-core Intel Xeon 3.73GHz (Netburst architecture) system with 2GB of RAM.

## 6. Conclusion and Future Work

We have presented a real time remote gesture visualization system for collocated shared interfaces. As the color transfer functions are estimated online, the lightning constraints are greatly relaxed. Online capture of the image desktop enables accurate detection whatever the desktop contains (static or moving windows, videos, etc.). Our system can be used with any kind of displays (plasmas for instance). However, detection of small object such as pens could be improved by further code optimization.

Our first main effort will be now to evaluate the influence of the remote gesture visualization on collaborative tasks. At the difference of most previous experiments [1][2] which are concerned with collaborative physical tasks on real 3D objects, we will focus on digital tasks suited to table interaction. As far as we have experimented, it appears that users are really enthousiastic with remote gesture visualization. It makes distant collaboration easy, natural and efficient because it mimics the collocated mutual awareness without any cognitive overload : users know who is doing what -or is intending to do so- at the distant site. Furthermore, users can precisely point out digital data to the distant users.

## Acknowledgments

This work was supported by the French government within the RNTL Project DigiTable and the Media and Networks Cluster.

## References

[1] D. Kirk, D. Stanton, and T. Rodden, "The effect of remote gesturing on distance instruction," *Proceedings of the Conference on Computer Support for Collaborative Learning (CSCL)*, 2005.

[2] R. Kraut, S. Fussel, and J. Siegel, "Visual information as a conversational resource in collaborative physical tasks," *Human Computer Interaction*, vol. 18, pp. 13–49, 2003.

[3] H. Ishii and M. Kobayashi, "ClearBoard: A Seamless Media for Shared Drawing and Conversation with Eye-Contact," *ACM SIGCHI*, pp. 525–532, 1992.

[4] J. C. Tang and S. L. Minneman, "Videodraw: a video interface for collaborative drawing," in *ACM Transactions on Information Systems*, vol. 9(2), pp. 170–184, 1991.

[5] N. Roussel, "Exploring new uses of video with videospace," in *Proceedings of the* 8th *IFIP International Conference on Engineering for Human-Computer Interaction (EHCI)*, vol. 2254, pp. 73–90, 2001.

[6] S. Malik and J. Laslo, "Visual touchpad: a two-handed gestural input device," *International Conference on Multimodal interface (ICMI)*, 2004.

[7] N. Takao, J. Shi, and S. Baker, "Tele-Graf¡ti: A Camera-Projector Based Remote Sketching System with Hand-Based User Interface and Automatic Session Summarization," *International Journal of Computer Vision*, vol. 53(2), pp. 115–133, 2003.

[8] C. Liao, M. Sun, R. Yang, and Z. Zhang, "Robust and Accurate Visual Echo Cancelation in a Full-duplex Projectorcamera System," *International Workshop on Projector-Camera Systems (ProCams)*, 2006.

[9] A. Tang, C. Neustaedter, and S. Greenberg, "Videoarms: embodiments in mixed presence groupware," in *Proceedings of the BCS-HCI British HCI Group Conference*, 2006.

[10] J. C. Tang and S. L. Minneman, "Videowhiteboard: video shadows to support remote collaboration," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 315 – 322, 1991.

[11] H. Ishii and M. Kobayashi, "Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments," *ACM Transactions on Information Systems*, vol. 11(4), pp. 349–375, 1993.

[12] D. Stotts, J. Smith, and D. Jen, "The Vis-a-Vid transparent facetop," in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*, pp. 57–58, 2003.

[13] K. Gyllstrom and D. Stotts, "Facetop: integrated semitransparent video for enahnced natural pointing in shared screen collaboration," (Chapel Hill, North Carolina), 2005.

[14] A. Wilson, "Touchlight : an imaging touch screen and display for gesture based interaction," *International Conference on Multimodal interface (ICMI)*, 2004.

[15] C. Dumas, S. Degrande, G. Saugis, C. Chaillou, M.-L. Viaud, and P. Pl´enacoste, "SpIn: a 3D Interface for Cooperative Work," *Virtual Reality Society Journal*, pp. 15–25, May 1999.

[16] H. Regenbrecht, T. Lum, P. Kohler, C. Ott, M. Wagner, W. Wilke, and E. Mueller, "Using Augmented Virtuality for Remote Collaboration," *Presence: Teleoperators and Virtual Environments*, vol. 13(3), pp. 338–354, 2004.

[17] H. Regenbrecht, M. Haller, J. Hauber, and M. Billinghurst, "Carpeno: Interfacing Remote Collaborative Virtual Environments with Table-Top Interaction ," *Virtual Reality*, vol. 10(2), pp. 95–107, 2006.

[18] F. Chen, P. Eades, J. Epps, S. Lichman, B. Close, P. Hutterer, M. Takatsuka, B. Thomas, and M. Wu, "ViCAT: Visualisation and Interaction on a Collaborative Access Table," in *Proceedings of the* 1st *IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, 2006.

[19] *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (Tabletop 2006), January 2006, Adelaide, Australia*, IEEE Computer Society, 2006.

[20] C. Shen, F. Vernier, C. Forlines, and M. Ringel, "Diamondspin: en extensible toolkit for around the table interaction," in *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 167–174, 2004.

[21] N. Roussel, "Experiences in the design of the well, a group communication device for teleconviviality," in *Proceedings of the* 10th *ACM International Conference on Multimedia*, pp. 146 – 152, 2002.

[22] P. H. Dietz and D. Leigh, "DiamondTouch: A Multi-User Touch Technology," in *ACM Symposium on User Interface Software and Technology (UIST)*, pp. 219 – 226, 2001.

[23] P. Huber, "Robust statistics," 1981.

[24] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric models," *Journal of Visual Communication and Image Representation*, vol. 6(4), 1995.