

Remote collaboration on desk-sized displays

By Mark Ashdown* and Peter Robinson



Work on remote collaboration has often focused on the person space created by a conventional videoconference where the participants see each other's faces, but we argue that a task space containing shared visual information is more important for most tasks. Trends in display technology mean that large visual task spaces can be created to maximize the shared context between collaborators. We have created a system called the Escritoire that presents users with a desk-sized projected display with bimanual input that allows documents and images to be arranged and modified by multiple remote collaborators. We describe the software architecture, the protocol that is used between the client and server programs, and the pen traces we have added to allow participants to gesture to each other in the large visual space to enrich their communication. Our user trials have shown that participants were able to use the system with a minimum of training, and found the traces useful in the collaborative setting. In future we will be connecting three or more sites together to explore the issues that arise with multi-party interaction on large shared desks.

Copyright © 2005 John Wiley & Sons, Ltd.

Received: 10 December 2003; Accepted: 21 April 2004

KEY WORDS: remote collaboration; large display; task space; person space

Visual Communication

We have developed a system to allow remote participants to interact in a graphical space where they share items such as documents and images. We argue that the shared context that this provides is more important for most tasks than the 'talking heads' model of a conventional video conference. We have created a display that is vastly bigger than a normal monitor. It fills a much bigger visual angle than a conventional display, and thus allows the user to stay aware of a much larger context of information, and to share a large working set of documents with a remote colleague.

In this first section we argue that a large visual task space is a desirable focus for remote collaboration, and in subsequent sections we describe the hardware configuration of the system we have built called the Escritoire, we describe how multiple such systems can be linked over the Internet to allow remote collaboration in

a large visual space, we describe the user interface of the Escritoire and the the user trials we conducted between remotely collaborating participants, and finally we give conclusions and directions for future work.

Task Spaces and Person Spaces

The Picturephone from Bell Labs was introduced publicly at the 1964 World Fair and at the time it was predicted to replace the existing voice-only telephone by the early 1970s. The intuitive appeal of video communication fuelled forecasts of its wide-scale adoption¹ but, except for limited use in business settings, it has not been widely used. A distinction can be drawn between a conventional videoconference that allows remote collaborators to be aware of each other's body positions, facial features and gaze directions, and one that allows them to share data such as text and images, and to interact with this data collaboratively. Bill Buxton uses the terms *person space* and *task space*² for the two types of interactive spaces that are created by these channels, which are depicted in Figure 1. Work on conferencing technology has generally focused on improving the fidelity of the video and audio links that support the 'talking heads' model, where the aim is to make it seem

*Correspondence to: Mark Ashdown, University of Cambridge Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK. E-mail: mark@ashdown.name

Contract/grant sponsor: Thales Research & Technology UK.

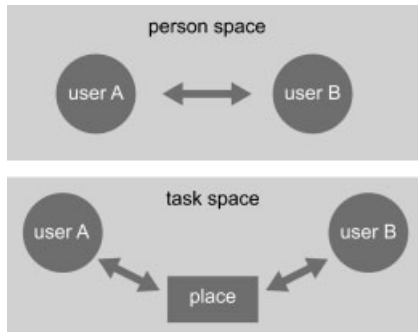


Figure 1. *Person space versus task space: a conventional videoconference creates a person space where the participants see each other; a task space is a new virtual space in which they collaborate.*

as if the participants are talking face to face. In contrast, technology to provide a task space creates a new virtual place to which all of the participants have access and where the interaction with data occurs.

Krueger's VIDEODESK,³ created around 1970, was an early example of a collaborative system that presented a task space to users at remote locations. Images of the users' hands were superimposed on the graphical data they were working on, so they could effortlessly talk and gesture to each other. Many previous studies have concentrated on the conventional video conference but some recent work has shown that the shared context of a task space is very helpful because it grounds users' conversation, which is especially important when they do not have words to describe the objects in the domain in which they are working.⁴

Interaction in a Task Space

In any kind of creative activity, the process, not just the end product, is important. After studying two-person design sessions, Bly⁵ concluded that the interactions on a drawing artifact are as important to many design collaborations as the final artifact itself. The conventional view of a shared drawing surface would be that it is merely a medium for creating and storing a drawing, but in studies of small group design sessions Tang⁶ found that approximately one third of the participants' actions were gestures—ephemeral actions meant to affect the flow of the session rather than to alter the shared design directly.

The designers of GroupSketch⁷ listed several criteria for cursors in group sketching software following their experience creating and using the system: all cursors

within a work surface should always be visible to all participants, cursors must have enough prominence on a multi-cursor display to attract the attention of other participants, cursor movements should appear continuously and with no apparent delay on all displays, and they should maintain the same relative location on every display so that they retain their relation to the work surface objects.

The creators of the Designer's Outpost⁸ added *transient ink* to their system in an attempt to allow users to make deictic gestures to each other. Annotations made in this mode faded away after a few seconds. About half of the users said they liked the feature, but they felt that having to activate it before using it disrupted the flow of ideas. A better method for gesturing is telepointer traces that continuously and unobtrusively show the recent positions of remote cursors, as implemented by Gutwin⁹ for remote participants working on a graphical task. Gutwin considered various visual representations of traces and favours simple motion lines that capture at most a second of motion, and fade from the start to the end of the trace. He has recently performed experiments that show that traces substantially increase the ability of users to recognize symbols like letters and numbers when they are created as gestures using a telepointer over a network that introduces jitter.¹⁰ In the experiments participants used a mouse to perform the gestures, which was sometimes difficult—it was suggested that a pen input device would have made it easier. After observing users of a standard shared whiteboard program, Wolf and Rhyne¹¹ asserted that freehand drawing with a mouse was difficult, but the ease of drawing with a pen makes hand-drawn marks a good solution for gesturing with a pen interface.

Large Displays

A conventional monitor, even a large one, only covers about 10% of a user's visual field, and about 1% of what one can see with a quick glance around. A simple, and increasingly common, way to employ more of the visual field using standard hardware is to add more monitors.¹² The conventional user interface has problems when it is applied to large monitors or multiple monitors, but some techniques have been developed to help: windows can be moved quickly by *bumping*¹³ them between monitors, and a *high-density cursor*¹⁴ fills the space between successive positions of the cursor to stop it getting lost on a large display. These are useful modifications to the conventional interface but the

long-term solution to lack of space is to use larger display devices.

The Xerox Liveboard¹⁵ was an early large display device. Tivoli,¹⁶ a whiteboard application created specifically for that display, prompted reconsideration of many tacit assumptions of traditional GUIs that indicated how a very large display is qualitatively as well as quantitatively different from a conventional monitor. A pen allows natural gestural input, does not require a cursor and generates direct position data, unlike the relative movements of a mouse that can be scaled to alter sensitivity. A large surface necessitates careful positioning of controls like menus, and places emphasis on keeping user control at the pen through techniques like gestural commands. Recent work at Microsoft Research focusing on large displays as the natural successor to conventional screens has shown that simply displaying the standard window system on a much larger screen can significantly increase productivity in realistic tasks.¹⁷

The Escritoire

We have created a system called the Escritoire¹⁸ that presents its user with a horizontal desk-sized display on which documents are displayed life-sized. The interface simulates sheets of paper on a real desk, by presenting sheets of virtual paper that can be arranged and annotated with bimanual input over the entire area of the desk.

Groups of several people at a single site often cooperate around information on paper,¹⁹ but the conventional monitor screen and interface devices are not well suited to this style of work because of their inherent design for private use by a single user. In contrast, the large projected display of the Escritoire allows people to gather around to cooperate, and information on such large displays has been shown to be assumed to be public by onlookers, who have no hesitation in reading it.²⁰ To allow participants at multiple sites to collaborate around items such as documents and images, two or more Escritoire desks can be linked, and combined with standard videoconferencing facilities. The desk display provides a large visual context that the users share, and they can collaborate in this task space, easily working together in a complex visual environment (Figure 2). Because of the large display surface and direct interaction with pens the collaborating users can quickly identify and move items around on the surface, and gesture to each other to complement their verbal communication.



Figure 2. We have linked two Escritoire desks to create a large shared visual space between participants containing document and images. They can also see and hear each other via a conventional videoconference.

Foveal Display

To make a display that is as large as a desk but also has high resolution where it is needed, we have used two overlapping projectors to create a *foveal display*, which has a large low-resolution region that encompasses a small high-resolution region to which items can be moved to view them in detail. Figure 3 shows how the projectors are arranged to create the horizontal display, where a projector below the desk creates a low-resolution *periphery* that fills the desk, while a projector above the desk with a shorter optical path to the surface creates a *fovea* that just covers an area of the desk close to the user. We compensate for the oblique projection and rough positioning of the projectors by using a short manual calibration procedure, then warping the projected graphics in real time using a commodity 3D graphics card.²¹

Baudisch *et al.* have also created a dual-resolution display, this time by combining an LCD panel with a surrounding projected display, and have shown that it improved speed and accuracy when extracting information from large static documents.²² Their system is a vertical screen that displays standard programs controlled with the keyboard and mouse, so it is much more like the conventional computer interface than the display of the Escritoire. We have also experimented with wall displays to complement the desk. A projector can create a very large, relatively low-resolution display on a wall which does not need very accurate control if it

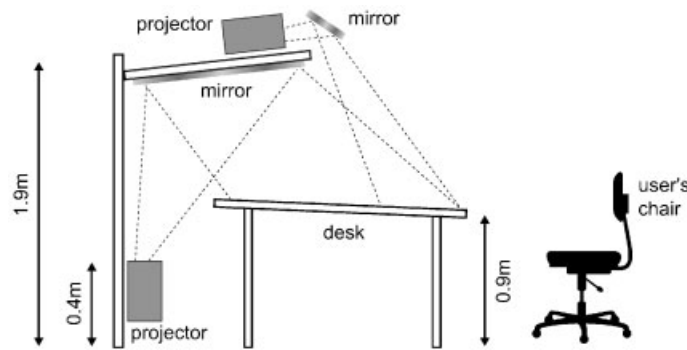


Figure 3. The arrangement of the projectors that create the foveal display of the Escrtoire. The bottom projector creates a display that fills the desk surface, while the top creates a higher-resolution region in front of the user.

is simply used to store documents that are currently not needed on the desk. We use a device whose position and orientation are tracked in three dimensions, and we control the display with the device by pointing at it from a distance. The wall display provides even more space than the large desk display and extends the concept of multiple display regions in which the size of a region is inversely related to its display resolution and the accuracy and frequency with which it is accessed.

Bimanual Input

To provide the types of interaction that are available on a normal desk we required bimanual (two-handed) input over the entire surface of the Escrtoire's desk-sized display, which we decided should be A0 size (1189×841 mm). However, no devices are currently available that support simultaneous two-handed input over a such a large surface. The Wacom Intuos tablets²³ are the only commercial devices we have found that sense two pens at once, but the maximum size of the tablets is only A3 (420×297 mm). Various large-format digitizers are available from GTCO Calcomp²⁴ in sizes exceeding A0, but they only allow input with a single pen.

Ultrasonically tracked whiteboard pens are available from eBeam²⁵ and Mimio.²⁶ A whiteboard marker pen is placed in a jacket which, when the pen is pressed down, emits an ultrasound signal that is received by microphones along the top of the board. These systems are designed to record marks that are made on a whiteboard, but a dummy pen can be placed in the jacket and the system can be used to control a computer display that is created by a projector. Laser scanning whiteboards are also available, such as the Webster LT Series.²⁷ A laser sweeps the region just above the whiteboard surface, identifying and locating the pen. Touch-sensitive boards are available which can be used either with a pen or with the fingers.²⁷⁻²⁹ They are generally constructed from two sheets of resistive material that are brought together when something touches the board. These touch-sensitive boards are not suitable for desk surfaces because users tend to lean on a desk as they use it which would create spurious readings.

To achieve simultaneous bimanual input over the whole desk we have used a GTCO Calcomp digitizer with a cordless pen as the desk, and have combined it with a Mimio (Figure 4). Because the digitizer uses electromagnetism for location and the other uses

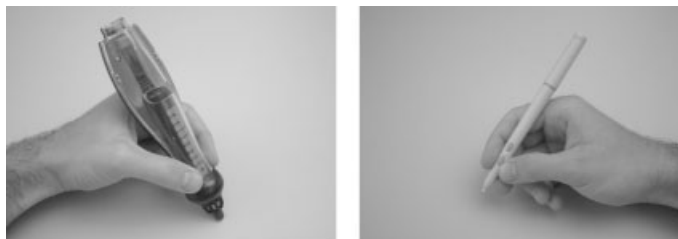


Figure 4. We have combined an ultrasonic pen for the non-dominant hand (left) with a electromagnetic digitizer pen for the dominant hand (right).

ultrasound they do not interfere with each other. We use the less accurate Mimio pen for the user's non-dominant hand, leaving the finer, more accurate digitizer pen for tasks such as writing with the dominant hand. An interesting alternative would be to make a desk that has an A3 Wacom Intuos tablet embedded in it, and also use the Mimio. Two Wacom pens could be put inside the Mimio jackets, which are designed to hold whiteboard markers, thus creating two pens that give high accuracy in the fovea, and lower accuracy in the periphery. The different accuracy levels would complement the different display resolutions in the two regions.

Two prototype technologies are being developed that allow multiple inputs to be tracked over a large horizontal surface, and could therefore be used to make this kind of interactive desk display in future. Researchers at Mitsubishi Electric Research Laboratories have created a system called DiamondTouch³⁰ that allows two users to interact simultaneously with a large surface using their fingers. A user cannot, however, use two hands simultaneously and independently, and cannot use a pen for precise work like writing. SmartSkin³¹ is a similar prototype that uses a mesh of copper wires under the surface of a table to detect the capacitance of users' hands. It can track multiple hands on a table-sized system and can estimate the distance of the hands from the surface. Anoto³² has an interesting technology that uses a camera embedded in a pen to track the movements of the pen as a person uses it to write on specially printed paper that has a pattern of tiny dots. A pen with Bluetooth does exist but currently there is no way to extract location data from it in real time—with this feature a cheap interactive surface could be created by simply creating the necessary patterns of dots on desk or wall surfaces; this would have the advantage that because the processing is contained within the pen rather than in the surface, many pens could be used on the same surface simultaneously.

Remote Collaboration

The dream of the paperless office has not been forthcoming, and paper is now used more than it ever was. Paper has many affordances that have not been surpassed by computer systems, such as the ability to lay it out over the large area of a desk so that multiple documents can be viewed at a glance and accessed by grabbing them. We have chosen to simulate paper in a computer system to address some of these issues. The

alternative would be to augment real paper with projected graphics as the DigitalDesk³³ and later systems did, but this approach has problems for remote collaboration because of the unavoidable asymmetry between the collaborating parties—the paper only exists at one site. The Designers Outpost,⁸ mentioned earlier, has this problem with paper notes that can be moved by local or remote users. Local users can move a note, but if a remote user wants to move one she must instead move its graphical representation, which then signals a *transactional inconsistency* to the local user, who must notice this and respond by moving the note to its correct new position.

Documents in PDF³⁴ form can be placed on the Escritoire's desk display, so electronic information can be exported from standard application programs just as it would be printed to paper, and existing paper documents can be scanned before being made available on the desk. The sheets of virtual paper on the desk can be put in piles to save space and to make fast informal groupings. This feature is like Apple's Pile Metaphor,³⁵ except that the piles do not contain icons representing the sheets, but the actual interactive sheets themselves.

We have augmented the video and audio channels of a conventional videoconference with interaction between two Escritoire desks to show how a task space can complement a person space. The shared desk surface creates a WYSIWIS (What You See Is What I See) system in which the participants' conversation can complement their work on the contents of the desk because each participant knows that they all see the same thing.

Client and Server

A WYSIWIS system in which the participants have a common view of shared information corresponds naturally to a client-server arrangement for the software, where the server holds the information and there is one client program for each participant. We have created a client and a server that have very different requirements and have thus been written in different styles. The client is a hardware-dependent program that handles the pen input devices and performs the real-time graphics warping using a commodity video card to compensate for oblique projection. It is designed to obtain maximum performance from the hardware. Control flow for the program is simple and uses two threads: one handles the input devices and sends corresponding messages to the server, and the other processes messages from the

	Client	Server
Control flow	Sequential	Event-driven
Programming language	C++	Java
System dependence	Dependent	Independent
State storage	Stateless	Stateful

Table 1. The different characteristics of the Escritoire's client and server programs

server and updates the display accordingly. The server program is very different: it is written in Java, making it system-independent, it provides the implementation behind all of the items such as PDF documents on the desk, and it stores all of the state so that the client can be restarted at any time. The division of labour between client and server is summarized in Table 1.

Figure 5 shows the general form of the software. To keep the client as simple as possible it represents the sheets of virtual paper as *tiles*: rectangular bitmapped images positioned on the desk with an ordering that determines how they occlude each other. The implementation of the sheets is handled by Java classes held in a tree on the server. The root of the tree represents the desk surface, and its children are the items on the desk, some of which may be piles which have children of their own (Figure 6). The state of the sheets that persists between sessions is saved on the server.

Using Java on the server makes it easy to write code for new types of objects to represent items on the desk, and we have written various types. A Desk object is always the root of the tree. It allows its children to be dragged around and put into piles. Otherwise it passes on all events to its children. A PDF object represents a PDF document, allows it to be annotated, and stores the strokes in the PDF file for later viewing offline. An Image object makes a bitmapped image available, and saves the annotations made to it. A Pile object groups sheets on the desk into a pile and has no visible form other than the arrangement of the sheets it contains. Pen movements and button presses are used to browse through items in a pile, reorder the items add items and remove items. We have also created a modified Java VNC client³⁶ which can be placed on the Escritoire and manipulated like any other sheet. This allows the displays of other computers running standard operating systems to be placed on the desk and controlled with the pen.

Protocol

The client and server exchange information using a specialized protocol that is implemented over TCP/IP. There are four types of message that are sent from client

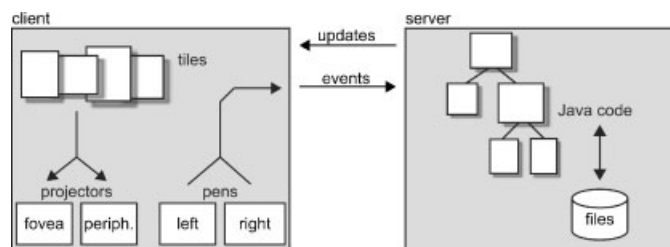


Figure 5. The client program handles the input and output devices: pens and projectors. The server maintains a tree of the Java objects that implement the sheets on the desk display, and stores the state associated with those sheets.

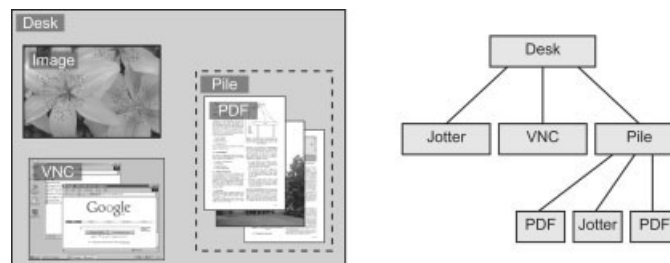


Figure 6. An example set of desk contents (left). The server stores the items as a tree of objects (right). The Desk object is always the root of the tree, Image and PDF objects store annotatable images and documents, a VNC object makes the display of another computer available, and a Pile object groups items.

to server: *hello* allows a client to connect to the server, and specifies the numbers of the displays to connect to because there may be multiple desk and wall displays; *keyboard* specifies a key code and whether the key was pressed or released; *pen* specifies which pen has changed its state, the number of the display it is on, and the state of the pen as x and y coordinates and an array for the pen buttons; and *ready* indicates that the client is ready to receive a burst of messages from the server. Changes in the pen state are used to generate events that are sent to the tree of Java objects. These events are analogous to the mouse events in standard window systems—pen moved, pen entered a sheet, pen exited a sheet, button pressed, button released, button clicked—except for the event to signal that the pen was lifted from the desk surface, which does not occur in a normal mouse-based system. Events are passed to the Desk object at the top of the tree, which either processes them or passes them down to the relevant child, which is chosen by testing the location of the pen event against the bounding boxes of the children.

There are eight types of message that are sent from server to client: *join* tells the client that it has successfully connected to the server; *create tile* causes a new tile to be created on the client and specifies a unique tile number, dimensions in metres and dimensions in pixels; *move tile* causes a tile to move to a new location; *update tile* updates a portion of a tile by specifying the coordinates of a pixel region and the pixel data itself; *destroy tile* removes a tile from the display; *order tiles* changes the z -order of the tiles by specifying an array of tile numbers; *cursor* moves a pen cursor by specifying a cursor number and location on the display; and *burst terminator* simply signals the end of a burst of messages from the server.

The Escritoire server outputs messages asynchronously because the Java code implementing the sheets on the desk display can start dedicated threads that can produce output at any time, and because multiple clients can connect and interact simultaneously. This suggests using a server-push scheme whereby as soon as a new message is created at the server it is transmitted to the client. However, the client has a large amount of graphics processing to do while updating the display, and forcing it to buffer messages while it is doing this is undesirable. Also, if the messages are buffered at the server there is potential for merging or discarding them, thus saving network capacity and processing at the client. This suggests using a client-pull system, where the client program polls the server for new messages, thus achieving an adaptive quality where the transfer

rate reduces as the client becomes more highly loaded. The solution we have employed is to switch between server-push and client-pull automatically. The message traffic between client and server is characterized by periods of inactivity when the user is thinking, punctuated by bursts of message data when the user is performing an action. During a burst of activity the client polls the server by sending it *ready* messages, and updates its displays with the new data, thus employing a client-pull scheme. The server responds to each of the client's *ready* messages with a burst of messages ended by a *burst terminator*. When the server has no more messages the burst contains only the *burst terminator*, and then the client switches to server-push mode. The server will then send a burst as soon as it has a new message.

The messages sent from server to client are buffered at the server, and merged when possible. This is most important for *update tile* messages because they contain large amounts of pixel data, so merging several messages into one can provide a large saving in transmission time. When an *update tile* message is buffered at the server the details of the tile region to be updated are stored, then when a similar message is buffered it is compared with each existing buffered region in turn: if the sum of the areas of the old region and the new region is greater than the area of their bounding box the two are merged because this will result in a saving.

User Interaction

We have designed the interface of the Escritoire to support remote collaboration. We added pen traces to allow remote collaborators to remain aware of each other's actions and to gesture to each other in the visual task space that they share. Note that the purpose of these gestures is to be understood by other humans, not by computers as in the case of gesture recognition. We tested the traces and various other features of the system in user trials between two separate sites.

Visual Feedback

We have implemented cursor traces like the ones by Gutwin mentioned earlier. The *cursor* messages from the server are saved with time stamps in a buffer, and used to create a trace of a certain period, which we have set at 0.7 seconds from experience. Gutwin reported that using transparency to draw a fading trace consumed a

large amount of processing power, so we have used various line thicknesses and dashed and dotted lines to achieve a fading effect that is very fast to draw with standard graphics functions (Figure 7 (right)).

Transient ink on the Designer's Outpost⁸ that was mentioned above was more like an annotation mechanism than support for gesturing, and one user of the system remarked that it was just as easy to make an annotation then erase it. Gutwin received favourable results for his pointer traces, which are more dynamic than the transient ink, but he used a mouse rather than a pen to make them, and anyone who has tried freehand drawing with a mouse will realize that this will restrict the accuracy with which precise gestures can be made. We believe that dynamic traces that do not need to be explicitly enabled before making a gesture provide good support for gesturing and awareness. Our results below show that they were favoured over simple cursors by remote collaborators.

User Trials

Initially we performed trials on a single-user system with seven people to get feedback on the interface. After a few minutes of instruction on using the system, participants were asked to highlight spelling mistakes in a series of short documents on the desk display, and were then asked to arrange sets of images in piles to test the piling feature. Encouragingly, all of the participants required only a few minutes of practice to be able to use the system. Of the three cursor options shown in Figure 7 participants generally preferred the first option: no cursor. This is not surprising, because unlike an indirect pointing device like a mouse, the direct pointing of the pen does not require a cursor to show the position, and if one is present it just gets in the way. Front projection

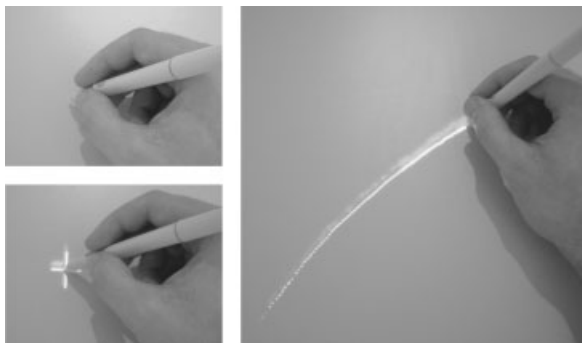


Figure 7. We have implemented three cursor options: no cursor, cross-hair, or trace.

usually has the problem that the display can be occluded by a person coming between the projector and the surface, but this has not been a problem on the Escritoire's desk display because the oblique projection from the back of the desk allows the user to lean forward a considerable distance without causing occlusion. Small occlusions are caused by a user's hands but no one mentioned this, possibly because people are used to having to move their hands when they occlude information and do so automatically.

To test the use of the Escritoire for remote collaboration we set three pairs of participants a realistic task for which they interacted through the desks and also through Microsoft NetMeeting using standard monitors and webcams, as shown in Figure 2. One desk was in Cambridge, UK, and the other was in Reading, UK, about 100 miles away. Initially each participant worked on her own: she was presented with 30 sheets on the desk, each containing details and a photograph of a house. She was told that the collaborative task would be to pick the best three houses from a group of them, and was instructed to look through the 30 houses and make any notes that might be useful on some blank sheets of virtual paper that were also on the desk. After the connection between the two sites was established each pair of participants performed a task three times: given 10 of the original 30 houses they found the best-value house, the second best and the third best. They used a different pen cursor each time—no cursors, cross-hairs and traces—in a different order for each pair. Each group of 10 houses took 20–30 minutes to discuss, and each complete trial took 2–3 hours. After completing the tasks the participants were asked which cursor type they preferred, and were asked for comments on the cursors. They were asked for a response to each of the following statements, on a Likert scale from 1 (strongly agree) to 5 (strongly disagree):

- The audio channel was useful for the task.
- The video channel was useful for the task.
- The desk interaction was useful for the task.
- The amount of desk space available aided the task.

The participants were then asked whether the difference in resolution between the fovea and periphery was a problem, whether the difference in brightness between the fovea and periphery was a problem, whether latency in the desk interaction was a problem, and finally they were asked for general comments about the system.

All of the participants understood the concept of interacting through the desk while conversing over a video conference link, and after they had used the desk

	Strongly agree	Agree	Neither	Disagree	Strongly disagree
Audio useful	6	0	0	0	0
Video useful	0	2	2	0	2
Desk useful	6	0	0	0	0
Space helped	3	2	1	0	0

Table 2. Responses from the six participants of the collaborative trials

system on their own they did not need any extra training to use it for distributed collaboration. One participant was split between preference for the cross-hair cursor and the trace, which made the totals 0 for no cursor, 0.5 for the cross-hair and 5.5 for the trace. This contrasts with the single-user trials, where the absence of a cursor was clearly preferred, so the participants did not want the traces for their own benefit but to allow better communication between participants. Without any cursor a participant had to wiggle a sheet to direct the attention of his collaborator to it; with the cross-hair he could indicate a single position on a sheet; and the traces were most useful by allowing a participant to direct his collaborators' attention precisely and by continually visualizing his actions.

Table 2 shows that the participants clearly thought the audio and desk channels were vital to the task but that the video was much less important. We believe that the task space of the desk will be more useful than the person space of the video channel in many tasks where participants share documents, and especially those where they share large amounts of visual information. The foveal regions of the two desks were in slightly different positions, which caused some problems when one participant could read the text of a document but the other could not. This would be remedied by using the intersection of all of the original foveal regions, which would waste pixels from the foveal projector, or setting a standard position for the fovea on all desks, which would constrain the positioning of the foveal projector. The difference in resolution between fovea and periphery was generally not considered to be a problem, and the difference in brightness was actually considered by some to be a benefit because it focuses attention on the foveal region and marks the boundary between low and high resolution.

Discussion

We undertook the collaborative trials described here with a server and client at two sites in the UK about 100

miles apart. They were linked over the Internet, with the server being connected via DSL, which is available to most UK homes, providing 256 kbps. This capacity was easily enough to carry the audio, video and desk channels during the main part of each trial—only small messages are required to update cursor and sheet positions, and to update a sheet when a user is annotating it. A period of a few minutes was taken at the start of each trial to download, from server to client, the bitmap data for the many items on the desk. This data was sent as raw bitmaps but we have since added PNG³⁷ compression to the software. This lossless image compression works very well for the PDF documents we have used, which are mostly comprised of text, giving a 92–98% saving. A more modest saving of 30–50% is achieved for photographs, so a system like JPEG may be necessary if users are to collaborate on tasks involving many natural images. The typical round trip time for packets sent over the Internet between the sites was around 25 ms, which is low enough to allow a remote user to interact fluidly with objects on the desk. However, the best-effort service provided by the Internet occasionally degraded for several seconds, presumably due to congestion in the network. Remote collaboration systems such as this one require latency guarantees if they are to ensure that participants can continuously interact with each other, just as latency in telephone systems must be limited to allow two callers to have a normal conversation. Continuing work on service-level specification standards indicates how requirements such as limits on latency might be specified to an Internet service provider. Performance management systems that support quality of service across multiple network domains are required to fulfil these specifications across end-to-end Internet connections, and the technology for such systems is now becoming available.³⁸

It would interesting to take the concept of display regions of different sizes and resolutions further by adding a projected wall display that could be used to store documents that are not currently in use. Two of the trial participants said they would have liked to have had private workspaces where they could hold material that

was not visible to the other person. A wall display could provide a private space for information that is not to be shared. Alternatively a user could use a laptop computer as a private space, moving documents across to the shared desk surface as necessary by dragging the files onto a client program that would transmit them to the server.

Projectors contain fans that generate noise, which must be considered when the audio hardware is being set up—careful positioning of the microphones will help to alleviate this problem. The video link could be improved by using a large screen to present a life-sized video image of the remote participant, which might allow a fairer comparison between person space and task space. We believe that the shared desk display is a good medium for collaboration because it allows users to share a large amount of information, and because it provides a common perspective to the users. Because of its WYSIWIS property and the fact that the users view and manipulate its contents from the same position, it supports the users' task of working together using the information.

The work on the *Escritoire* system that we have described here has been undertaken at the University of Cambridge Computer Laboratory in conjunction with Thales Research & Technology UK (www.thalesresearch.com). Starting in 2004 we will be performing further work on remote collaboration in large visual task spaces, and will be working with more partners including the Decis Lab in the Netherlands (www.decis.nl) which includes Thales Research & Technology Netherlands. Multiple desk displays providing shared access to documents and images will allow remote collaboration on proposals, plans, progress reports and results, and an aim of the project is to support interaction between the various European partners and to reduce the need for travel between sites. The *Escritoire* system has been designed to allow many desks to be linked together but we have not yet had the hardware to try this out. We expect that linking three or more desks will prompt new insights and developments.

ACKNOWLEDGEMENTS

Thales Research & Technology UK funded this work and have helped with the construction of the hardware and the user trials. Their involvement is gratefully acknowledged.

References

1. Egido C. Video conferencing as a technology to support group work: a review of its failures. In *Proceedings of CSCW 88*, 1988; pp. 13–24.

2. Buxton W. Telepresence: integrating shared task and person spaces. In *Proceedings of Graphics Interface '92* 1992; pp. 123–129.
3. Krueger MW. Environmental technology: making the real world virtual. *Communications of the ACM* 1993; 36(7): 36–37.
4. Kraut RK, Gergle D, Fussell SR. The use of visual information in shared visual spaces: informing the development of virtual co-presence. In *Proceedings of CSCW 2002*, 2002; pp. 31–40.
5. Bly SA. A use of drawing surfaces in different collaborative settings. In *Proceedings of CSCW 88*, 1988; pp. 250–256.
6. Tang JC, Leifer LJ. A framework for understanding the workspace activity of design teams. In *Proceedings of CSCW 88*, 1988; pp. 244–249.
7. Hayne S, Pendergast M, Greenberg S. Gesturing through cursors: implementing multiple pointers in group support systems. In *Groupware for Real-Time Drawing: A Designer's Guide*, Greenberg S, Hayne S, Rada R (eds). McGraw-Hill: London, 1993; 63–80.
8. Everitt KM, Klemmer SR, Lee R, Landay JA. Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration. In *Proceedings of CHI 2003*, 2003; pp. 553–560. Also available as University of California, Berkeley, Technical Report UCB//CSD-02-1201.
9. Gutwin C. Traces: visualizing the immediate past to support group interaction. In *Proceedings of Graphics Interface 2002*, 2002; pp. 43–50.
10. Gutwin C, Penner R. Improving interpretation of remote gestures with telepointer traces. In *Proceedings of CSCW 2002*, 2002; pp. 49–57.
11. Wolf CG, Rhyne JR. Gesturing with shared drawing tools. In *Conference Companion for InterCHI 93*, 1993; pp. 137–138.
12. Grudin J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proceedings of CHI 2001*, 2001; pp. 458–465.
13. Regan T, Czerwinski M, Meyers B, Smith G. Bumping windows between monitors. *Tech. Rep. MSR-TR-2003-13*, Microsoft Research, 2003.
14. Baudisch P, Cutrell E, Robertson G. High-density cursor: a visualization technique that helps users keep track of fast-moving mouse cursors. In *Proceedings of Ninth IFIP International Conference on Human-Computer Interaction (INTERACT 2003)*, 2003.
15. Elrod S, Bruce R, Gold R, Goldberg D, Halasz F, Janssen W, Lee D, McCall K, Pedersen E, Pier K, Tang J, Welch B. Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI 92*, 1992; pp. 599–607.
16. Pedersen E, McCall K, Moran TP, Halasz FG. Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proceedings of InterCHI 93*, 1993; pp. 391–398.
17. Czerwinski M, Smith G, Regan T, Meyers B, Robertson G, Starkweather G. Toward characterizing the productivity benefits of very large displays. In *Proceedings of IFIP Interact 2003*, 2003; pp. 9–16.
18. Ashdown M, Robinson P. A life-sized desk display for peripheral awareness and remote collaboration. In *First Research Workshop on Augmented Virtual Reality (AVIR 2003)*, 2003.

19. Luff P, Heath C, Greatbatch D. Tasks-in-interaction: paper and screen based documentation in collaborative activity. In *Proceedings of CSCW 92*, 1992; pp. 163–170.
20. Tan D, Czerwinski M. Examining HCI attributes of peripheral and physically large displays. In *CHI 2003 Workshop on Providing Elegant Peripheral Awareness*, 2003.
21. Ashdown M, Robinson P. Experiences implementing and using personal projected displays. In *IEEE International Workshop on Projector–Camera Systems (Procams 2003)*, 2003.
22. Baudisch P, Good N, Stewart P. Focus plus context screens: combining display technology with visualization techniques. In *Proceedings of UIST 2001*, 2001; pp. 31–40.
23. Wacom. <http://www.wacom.com> [20 April 2004].
24. GTCO Calcomp. <http://www.gtco.com> [20 April 2004]. This company makes various tablets and digitizers for different applications.
25. eBeam. <http://www.e-beam.com> [20 April 2004]. This company makes an ultrasonic pen-tracking system.
26. Mimio from virtual ink. <http://www.mimio.com> [20 April 2003]. An ultrasonic pen input device that can be attached to whiteboards.
27. Webster. <http://www.websterboards.com/> [20 April 2004].
28. SMART Technologies Inc. <http://www.smarttech.com> [20 April 2004]. This company makes the SMART Board, a touch-sensitive whiteboard.
29. TeamBoard. <http://www.teamboard.com> [20 April 2004]. This company makes a touch-sensitive whiteboard.
30. Dietz P, Leigh D. DiamondTouch: a multi-user touch technology. In *Proceedings of UIST 2001*, 2001; pp. 219–226.
31. Rekimoto J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of CHI 2002*, 2002; pp. 113–120.
32. Anoto. <http://www.anoto.com> [20 April 2004].
33. Wellner PD. Interacting with paper on the DigitalDesk. *Communications of the ACM* 1993; **36**(7): 87–97.
34. Adobe Portable Document Format (PDF). <http://www.adobe.com/products/acrobat/adobepdf.html> [20 April 2004].
35. Mander R, Salomon G, Wong YY. A ‘pile’ metaphor for supporting casual organization of information. In *Proceedings of CHI 92*, 1992; pp. 627–634.
36. Virtual network computing (VNC), a remote display system. <http://www.realvnc.com> [20 April 2004].
37. Portable network graphics. <http://www.libpng.org/> [20 April 2004].
38. Trimintzios P, Andrikopoulos I, Pavlou G, Flegkas P, Griffin D, Georgatos P, Goderis D, T’Joens Y, Georgiadis L, Jacquenet C, Egan R. A management and control architecture for providing IP differentiated services in MPLS-based networks. *IEEE Communications Magazine* 2001; 80–88.

Authors’ biographies:



Mark Ashdown has recently finished a PhD in Computer Science at the University of Cambridge Computer Laboratory, where he previously obtained a BA in Computer Science. His PhD work has focused on large projected interfaces, calibration for such displays including the use of computer vision, input devices, graphics and human–computer interaction. In 2004 he will be starting a post-doctoral position at the University of Tokyo, Japan.



Peter Robinson is Reader in Computer Technology and Deputy Head of Department at the University of Cambridge Computer Laboratory in England. His research interests are in the general area of applied computer science. The main focus for this is human–computer interaction, where he has been leading work for some years on the use of video and paper as part of the user interface. He also works on electronic design automation and, in particular, on support for self-timed circuits.