

Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments

Jun Rekimoto

Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda, Shinagawa-ku,
Tokyo 141-0022 Japan
Phone: +81 3 5448 4380
Fax: +81 3 5448 4273
E-Mail: rekimoto@acm.org
<http://www.csl.sony.co.jp/person/rekimoto.html>

Masanori Saitoh

Department of Computer Science,
Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama, Kanagawa 223 Japan
saitoh@aa.cs.keio.ac.jp

ABSTRACT

This paper describes our design and implementation of a computer augmented environment that allows users to smoothly interchange digital information among their portable computers, table and wall displays, and other physical objects. Supported by a camera-based object recognition system, users can easily integrate their portable computers with the pre-installed ones in the environment. Users can use displays projected on tables and walls as a spatially continuous extension of their portable computers. Using an interaction technique called hyperdragging, users can transfer information from one computer to another, by only knowing the physical relationship between them. We also provide a mechanism for attaching digital data to physical objects, such as a videotape or a document folder, to link physical and digital spaces.

KEYWORDS: multiple device user interfaces, table-sized displays, wall-sized displays, portable computers, ubiquitous computing, architectural media, physical space, augmented reality

INTRODUCTION

These days people can take small yet powerful computers anywhere at anytime. Modern notebook-sized portable computers have of several gigabytes of disk storage, processing power almost equal to desktop computers, and an integrated set of interface devices (LCD screen, keyboard, and pointing device). Therefore, it is not impossible to store and carry almost all one's personal data (documents, presentation slides, or digital images) in such a small computer.

In parallel with this tendency, our working environments, such as meeting rooms, are going to be equipped with many computing facilities such as data projectors and digital

whiteboards. It is becoming quite common during a meeting to make a presentation using a video projector to show slide data stored in the presenter's portable computer. It is also very common for meeting attendees to bring their own computers to take notes. In the near future, we also expect that meeting room tables and walls will act as computer displays. Eventually, virtually all the surfaces of the architectural space will function as computer displays [8]. As Lange et al. [5] pointed out, large and multiple display surfaces are essential for supporting collaborative, or even individual, activities. We can simultaneously spread several data items out on these surfaces without hiding each other.

Considering these two trends, the natural consequence would be to support smooth integration between portable/personal and pre-installed/public computers. However, in today's computerized meeting rooms, we are often frustrated by poor supports for information exchange among personal and pre-installed computers. In our physical lives, it is quite easy to circulate physical documents among meeting participants and spread paper diagrams on the table, or hang them on the wall. During a meeting, participants around the table can quickly re-arrange these diagrams. When they are displayed on computer screens, information exchanges between computers often require tedious network settings or re-connection of computers. It is not easy to add annotations to an image on the projector screen while another participant is presenting his data on that screen. When you want to transfer data from your computer to others', you might need to know the network address of the target computer, even if you can physically identify that computer.

In this paper we describe our design and implementation of a computer augmented environment that allows a user to smoothly interchange digital information between their portable computers and a computerized table and wall. Using the combination of camera-based marker recognition and interaction techniques called hyperdragging and anchored cursors, users can easily add their own portable computers to that environment. This intuitive, easy-to-use system is just like dragging icons from on screens to another in a single

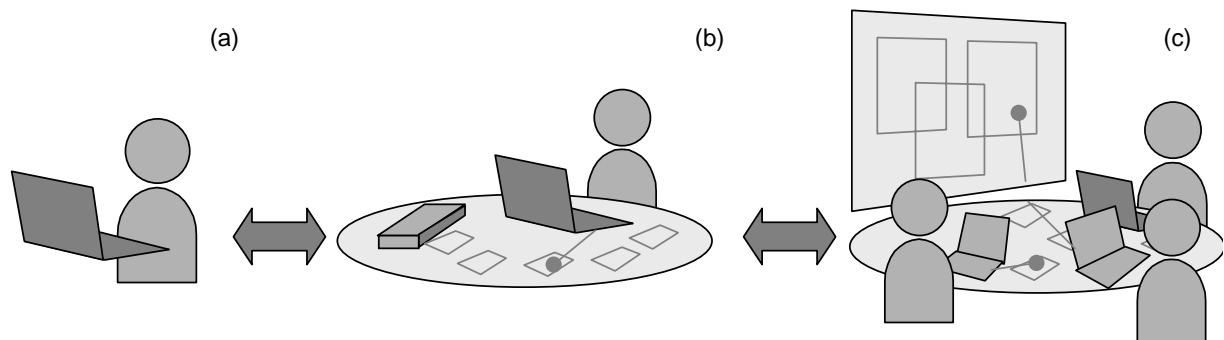


Figure 1: Evolution of spatially continuous workspaces: (a) A user can perform individual tasks with a portable computer. (b) The table becomes an extension of the portable computer. (c) Pre-installed computer displays (table and wall) also serve as shared workspaces for collaborative tasks.

computer supports multiple monitors. People can move information between different computers by only using normal mouse operations and only knowing the physical relationship among them. The system also provides a mechanism for attaching digital data to physical objects, such as a videotape or a document folder, to make tight connections between physical and digital spaces.

A SPATIALLY CONTINUOUS WORKSPACE

While many research systems on augmented physical spaces use pre-installed computers for interaction, we are more interested in how we can smoothly integrate our existing portable computers with the pre-installed ones.

The key features of our system design can be summarized as follows:

Environmental computers as extensions of individual computers

In our design, users can bring their own portable (notebook or palmtop) computers into the environment and put them on the table. Then, the table becomes an extended desktop for the portable computers (Figure 1). That is, the user can transfer digital objects or application windows to the displays on table/wall surfaces. They can use a virtually bigger workspace around the portable computer.

The user manipulates digital objects on the table (or on the wall) using the input devices (such as a track-ball or a keyboard) belonging to the portable computer. Instead of introducing other interaction techniques such as hand-gesture recognition, we prefer to use portable computers because notebook computers already have an integrated set of interaction devices that are enough for most applications. With these interaction devices, users do not have to change user-interface style while dealing with the table or wall. In addition, many recent sub-notebook computers have audio I/O devices, so they can also be used to create voice notes during the task.

If two or more users sit at the same table, the table also becomes a shared workspace among them; the participants can freely interchange information among the participating

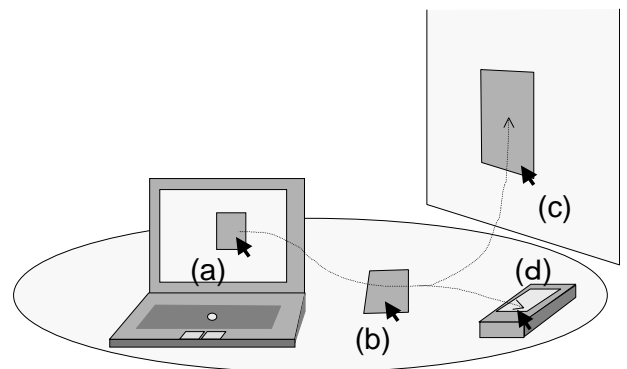


Figure 2: Hyperdragging: A spatially continuous interaction technique for moving information between computers. (a) A user can start moving an object on a computer in the normal manner by dragging it with the pointing device. (b) When the cursor reaches the edge of the screen, it "jumps" to the table surface. (c) The user can continue to drag it to another surface, such as a wall. (d) The user can also drop an item on a physical object, such as a VCR tape, to make a link between real and virtual objects.

portable computers by placing information items on the table/wall.

Support for links between digital information and physical objects

In addition to providing support for portable computers, the system allows users to put non-electronic objects such as VCR tapes or printed documents on the table. By reading an attached visual marker on the object, the system recognizes it and displays digital data that is linked to that object. The user can also add other digital information by simply dragging-and-dropping it onto the object.

Although other systems also support links between physical and digital objects (such as InfoBinder[15], mediaBlocks[18], and Passage[7]), these objects are only for carrying digital data and there are no particular roles in a real world. On the other hand, we are more interested in making a link between digital contents and things *that also have specific roles in the real world*. For example, we can attach editorial instructions



Figure 3: A meeting with InfoTable and InfoWall

to a VCR tape, as a digital voice note. We can also bind physical documents and digital data in a single document folder.

Spatially Continuous Operations

During these operations, we pay special attention to how the physical layout of objects (computers and other real objects) can match the digital manipulations. In other words, the user can use the integrated spatial metaphor for manipulating information in the notebooks, on the table or wall surfaces, and other physical objects placed on the table (Figure 2). For example, when the user wants to transfer data from a notebook computer to the table, he/she can simply drag it from the notebook screen to the table surface across the boundary of the notebooks. At the edge of the notebook screen, the cursor automatically moves from notebook to table. The user can also attach digital data to the physical object by simply dragging and dropping it onto the physical object.

INFOTABLE and INFOWALL: A PROTOTYPE HYBRID ENVIRONMENT

To explore the proposed workspace model, we developed a computer-augmented environment consisting of a table (called InfoTable) and a wall (called InfoWall) that can display digital data through LCD projectors. Figure 3 shows the system configuration of our environment. In this environment, users can dynamically connect their portable computers to perform collaborative and individual tasks. This section summarizes the user-interface features of the system.

We make some assumptions about the portable computers that can be integrated into the environment. To enable the portable computers to be identified by the pre-installed environmental computers, we attach a small visual marker (printed 2D barcode) to each portable computers and other physical object. Portable computers are also equipped with a wireless network for communicating with other computers.

Hyperdragging

When a user sits at the table and puts his/her portable computer on the table, a video camera mounted above the table finds its attached visual marker and identifies the owner of the computer. At the same time, the location of the computer is also recognized.

When the user wishes to show his/her own data to other participants, he/she can use an interaction technique called hyperdragging (Figure 4). That is, the user presses the mouse cursor on a displayed item and drags it toward the edge of the computer screen. When the cursor reaches the edge of the display, it migrates from the portable computer to the table



Figure 4: Moving information using "hyperdragging": A user can drag-and-drop a digital object between a notebook PC and a table surface display. During its operation, an "anchored cursor" line connecting the cursor and the notebook appears on the table display.



Figure 5: The anchored cursor shows the link between information on the table and the notebook computer

surface (Figure 4, middle). If the cursor is grabbing an object, the dragged object also migrates from the portable computer to the table surface. By manipulating the cursor, the user can place the object at any location on the table. Furthermore, the user can move the item toward the edge of the table, to cause a hyperdrag between the InfoTable and the nearby InfoWall display (Figure 4, bottom panel).

This hyperdragging technique supports the metaphor of the table being a spatially continuous extended workspace for portable computers. Users can place data items such as text or graphics around the notebook computer, as if they had a virtually bigger computer desktop.

The combination of two different displays -- a high-resolution small display on the portable computer and a low-resolution large display on the table -- represents the user's focal and peripheral information space. While keeping the focal objects on the notebook screen, the user can spread a number of items around the computer. When the user needs one of them, he/she can immediately hyperdrag it back to the notebook screen.

Anchored cursor

While a user is manipulating his/her cursor outside the notebook computer, a line is projected from the portable computer to the cursor position. This visual feedback is called the *anchored cursor*. When multiple users are simultaneously manipulating objects, there are multiple cursors on the table/wall. This visual feedback makes it easy for all participants to distinguish the owner of the cursors. When two or more participants manipulating objects on the table or on the wall, anchored cursors indicate the owner of the cursor in a visual and spatial way.

The anchored cursor is also used to indicate the semantic relationships between different display surfaces. For example, while the user navigates through a large map projected on the

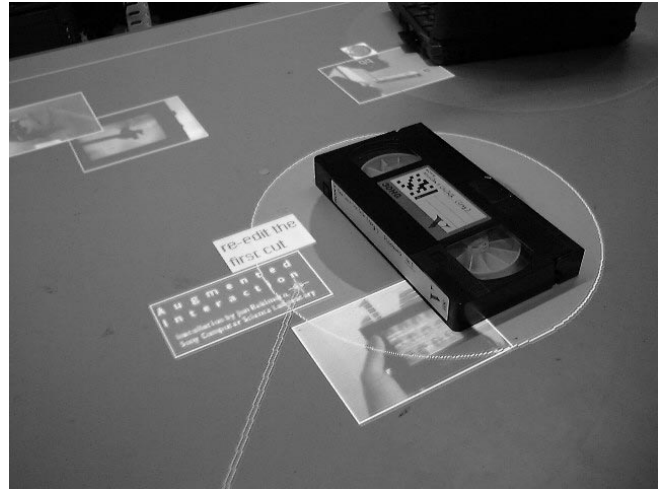


Figure 6: A recognized object (a VCR tape) with an "object aura": A user can attach a digital item by dropping it onto the object aura.

table, a notebook computer continuously displays detailed information related to the current cursor position (Figure 5). The anchored cursor shows the visual connection between them.

Table and wall as shared information surfaces

The InfoTable/InfoWall surfaces can also act as an integrated shared information space among participants. When two or more users sit at the InfoTable, they can freely place data objects on the table from their notebook computers.

Unlike desktop computer's screens, or augmented desk systems [22], there is no absolute notion of the "top" or "bottom" of the screen for table-type computers. Thus the multi-user capability of the InfoTable causes interesting user-interface design issues for determining the above sides. InfoTable uses the recognized spatial position of notebook computers to determine which is the "near" side for each user. For example, when a user brings a diagram from the far side to the near side of the user, the system automatically rotates it so that the user can read it.

Object aura

The system also supports the binding of physical objects and digital data. When an object (such as a VCR tape) with a printed visual marker is placed on the InfoTable, the system recognizes it and an oval-shaped area is displayed at the location of that object. This area, called the "object aura", representing the object's information field (Figure 6). This visual feedback also indicates that the physical object has been correctly recognized by the system.

The object aura represents a data space for the corresponding object. The user can freely attach digital data, by hyperdragging an object from the table surface and dropping it on the object aura. For example, if the user wants to attach a voice memo to the VCR tape, he/she first creates a voice note on his/her notebook computer (using its built-in microphone),

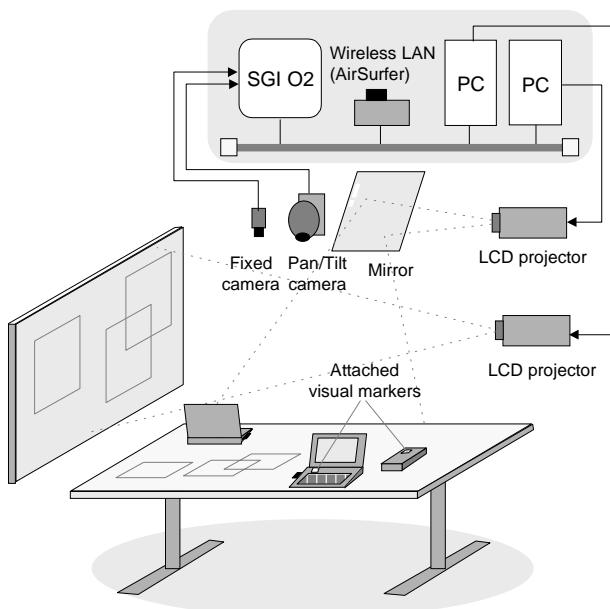


Figure 7: System configuration

and then hyperdrags it from the notebook screen to the VCR tape's aura. When the user releases the mouse button, the voice note is linked to the VCR tape. When someone physically removes the object from the table, the attached data is saved in the network server. This data is re-displayed when the object is placed on any InfoTable.

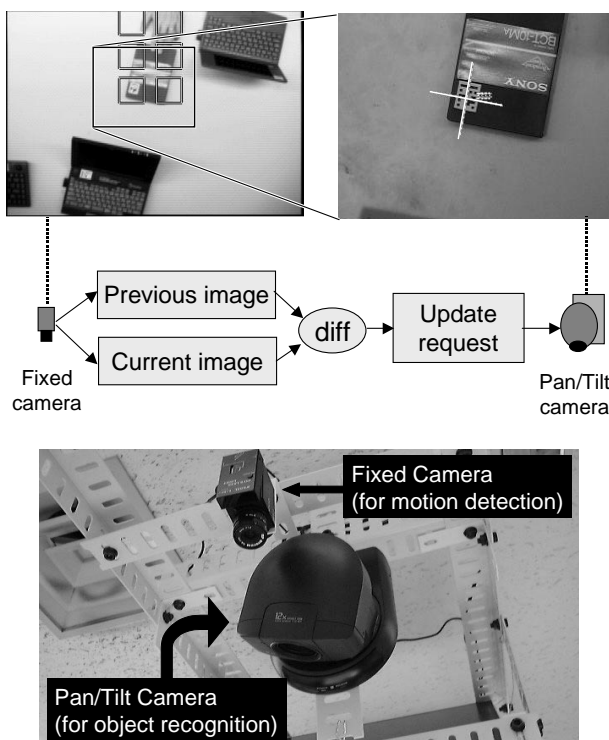


Figure 8: DeskSat uses a combination of two cameras for object recognition

SYSTEM ARCHITECTURE

To enable the interactions described in the previous section, we installed a computer projector and a set of CCD cameras (about 160 cm) above the table. Beside the table, we also installed the combination of a whiteboard and another computer projector as a wall-sized display. Figure 7 shows the device configuration of the system.

Desksat

For the video camera used as an object recognition sensor, there is a tradeoff between camera resolution and the field of view. The camera resolution must be high enough to identify fairly small visual markers that are attached on objects. High-resolution images should also be useful for making a record of the table. However, currently-available video cameras do not cover the entire table surface with the required high resolution. DigitalDesk [22] attempted to solve this problem by adding a second video camera, which is used to capture a fixed sub-part of the desk with higher resolution than the first one. A user is guided to place a document on that focal area.

Our solution is to use a combination of two cameras (Figure 8). The first one is a motor-controlled video camera (Sony EVI-D30) that changes its panning, tilting, and zooming parameters according to commands from the computer. This camera can capture the entire table surface as well as a part of the area with higher resolution (up to 120 dpi) when the camera is zoomed in. Normally, this pan/tilt camera is scanning over the surface of the table by periodically changing the direction and orientation of the camera head. We divided the table surface into a 6-by-6 mesh and the pan/tilt camera is controlled to regularly visit all 36 areas. We called this scheme "Desksat", by analogy to Landsat (land-satellite). In our current setup, it takes about 30 seconds to visit all the areas, including camera control and image processing (marker recognition) times.

The second camera is a fixed camera that is always looking at the entire table surface. This camera analyzes changes on the table from the difference between video images. Then it determines which sub-area has been changed and sends an "area changed" event to the pan/tilt camera. Using this event information, the pan/tilt camera can quickly re-visit the changed area. We choose a threshold value for difference detection so that the fixed camera is not affected by the projected image.

We use a small amount of heuristics to determine the order of visiting these changed areas. Since people normally use the table from the outside, changes in the inner areas are more likely to be object changes. Thus we assign higher priorities to inner areas than to outer areas; when the fixed camera finds several changes simultaneously, the pan/tilt camera checks these areas from inside to outside.

Using these techniques, when a user puts, moves (or removes) objects on the table, this effect will be recognized



Figure 9: Visual marker recognition and obtained position and orientation.

by the system within a few seconds. Although this response time might not be satisfactory for applications that require continuous/realtime object tracking, such as the one in [20], this scheme suits our circumstances quite well where changes occur only intermittently.

Visual marker recognition

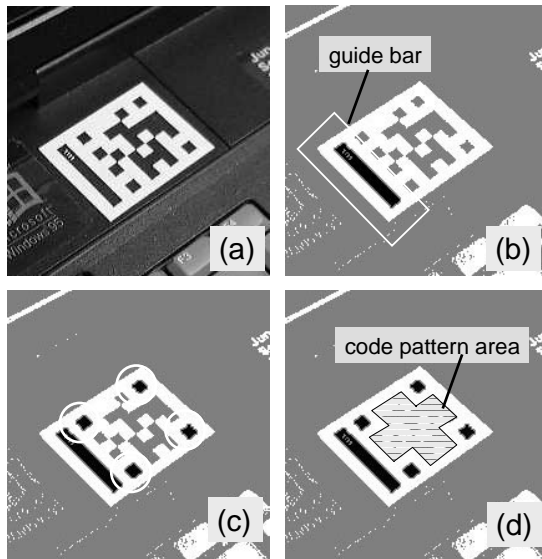


Figure 10: The visual marker recognition algorithm: (a) Original image. (b) Binarized image. Connected regions that have the specific second-order moment are selected. These regions become candidates of a guide bar of the marker. (c) Four corners of the marker region are searched based on the guide bar position/orientation. (d) If the guide bar and the four corners are successfully found, the system finally decodes the bitmap pattern in the marker. Based on the corner positions of the marker, the system estimates and compensates for the distortion effect caused by camera/object tilting. Then the system decodes the code bit pattern. After checking for the error bits, the system determines whether or not the image contains a correct 2D marker.

The printed visual markers (2D matrix code) attached to objects (including portable computers and other non-electronic objects) on the table can identify 2^{24} different objects using the combination of printed matrix patterns (we use a slightly different version of the matrix code system described in [10]). Using the Desksat architecture described above, 2D markers as small as $2\text{cm} \times 2\text{cm}$ can be recognized from the pan/tilt camera above the table.

In addition to its ID being recognized, the marker's position and orientation are also identified (Figure 9). This information is used to calculate object positions in related to the marker position. For example, the position of the cursor on the table while the user is doing a hyperdrag, is calculated based on the current position/orientation of the marker attached on the portable computer. The marker recognition algorithm is summarized in Figure 10.

Since 2D codes cost virtually nothing and can be printed, there are some uses that could not be achieved by other ID systems. For example, we can use small Post-it notes with a 2D code. This (physical) Post-it can convey digital data such as voice notes or photographs with an attached ID.

Hyperdragging

To enable hyperdragging (when the user moves the cursor of the notebook computer from notebook to the table), the system designates mouse-sensitive areas along all four edges of the notebook screen. When the cursor enters this area, the system re-maps the cursor position to the screen, and calculates the offset of this remapping to maintain the cursor position on the table. While the real (original) cursor stays near the edge of the notebook screen, the user can control the virtual cursor position on the table by continuing to press the pointing device.

To correctly calculate the cursor position on the table, the system also has to know the notebook's position and orientation on the table. The system gets this information from an attached visual marker on the notebook PC. Figure 9 shows how the system finds the PC position/orientation based on the attached marker.

Object migration

As a result of hyperdragging, the system needs to transfer data between two computers (e.g., from a notebook computer to the computer running the table display). All application programs for our environment are written in Java and the system employs Java's object serialization mechanism and the remote method invocation (RMI) method to transfer objects. Currently we support text, sound (voice notes), URLs, file short-cuts, and image data as migratable object classes.

EXPERIENCE AND DISCUSSIONS

Up to the time this paper was written, no formal evaluation had been conducted. However, with this environment, the authors and their colleagues in the laboratory have experimentally tried several collaborative activities including a

group meeting.

The concept of hyperdragging was instantly understood by the users and well accepted. Many users were surprised that they could freely move objects between different computers and other physical objects, with a simple drag-and-drop operation. People also appreciated being able to attach data onto the wall surface while sitting at the table. Many wished that they could also move physical objects with the cursor! Anchored cursors were also helpful when two or more users were performing operation simultaneously, especially when the users manipulated object far from their positions. Some users suggested (and we are considering implementing) putting small peripheral devices, such as printers or scanners, on the table and supporting hyperdragging to them. For example, the user could drop an image object onto the printer for making a hardcopy of it.

Some users felt that moving an object across a larger distance was tiresome. We might be able to incorporate techniques other than dragging, such as described in[2]. We also felt that the mapping scale between pointer movement and the pointing device greatly affects usability. Since the projector resolution on the table (about 20 dpi) is much coarser than the notebook computer's (100-110 dpi), mapping without scaling causes a discontinuous change in cursor speed at the boundary between the notebook and the table.

We also observed that there were interesting differences between hyperdragging and our previous multi-device interaction technique called "pick-and-drop"[9, 11]. Pick-and-drop uses a digitizer stylus to pick up a displayed object from one screen and drop it on another screen. Pick-and-drop is a more direct and physical metaphor than hyperdragging, because its operation is quite similar to picking up a real object. Hyperdragging allows a user to manipulate objects that are out of the user's physical reach, while pick-and-drop does not. Pick-and-drop requires a stylus-sensitive surface for operation, but hyperdragging works on any display and projected surfaces.

There is also the question of suitability between pointing devices and interaction styles. Apparently pick-and-drop is best suited for a pen, while hyperdragging does not work well with a pen because it forces indirect mapping between the pen position and the cursor position. On the other hand, hyperdragging is more suitable for a track-ball or a track-point, and these are common for notebook-sized computers.

RELATED WORK

Research on augmenting face-to-face interactions often assumes pre-installed computer facilities so the configuration of computers is fixed. For example, Colab[17] provides a projector screen and table-mounted computers for participants. There was no support for incorporating other computers that the participants might bring to that environment. However, considering recent trends in mobile computing, it would be more practical to support dynamic connections between

mobile and pre-installed computers.

There are several systems that project digital information onto the surface of a physical desk. VIDEODESK[4] consists of a light table and a video camera. The user can interact with the other participant's silhouette projected onto the table. DigitalDesk [21, 22] allows interactions between printed documents and digital information projected on a desk. A recent version of the DigitalDesk series also added a document identification capability based on OCR[13]. Luminous Room[19] (and its underlying "I/O bulb" concept) uses a video projector mounted on a computer-controlled gimbal to change the projection area. Its application called Illuminating Lights[19] helps a holography designer to rapidly layout physical optics devices on the desk. Streitz et al. developed a set of computer augmented elements including a wall, chairs, and a table[7]. Among them, the InteracTable is a table-sized computer supporting discussion by people around it. It also displays information which is carried by a physical block called "Passage". While these systems mainly focus on interaction between non-electronic objects and projected digital information, our system also supports information interchange among portable computers, table/wall surfaces, and physical objects.

The Desksat architecture was partially inspired by the whiteboard scanning system called ZombieBoard[14]. Zombieboard controls a pan/tile camera to capture the mosaic of partial whiteboard images. By joining these images together, a higher resolution image of the entire whiteboard can be produced. The Brightboard [16] is another example of a camera augmented whiteboard system; it recognizes hand-drawn commands made by a marking pen.

As for multi-computer interactions, the Hybrid User Interfaces [1] is an application for a see-through head-mounted display that produces a virtually bigger screen around the screen of the desktop computers. The PDA-ITV system[12] uses a palmtop computer (Apple Newton) as a commander for an interactive TV system. These systems assume a fixed-devices configuration, and are mainly designed for single-user applications.

Ariel [6] and transBOARD[3] support connections between barcode-printed documents or cards and digital contents. Insight Lab[5] is a computer supported meeting room that extensively uses barcoded tags as physical/digital links and commands. These systems normally require a manual "scan" of each printed barcode. This may become a burden for users, especially when they have to deal with a number of barcodes. These systems do not recognize the location of each object, so they require other mechanism to achieve spatially continuous operations.

CONCLUSIONS AND FUTURE WORK

We have described our design and implementation of a hybrid work space, where people can freely display, move, or attach digital data among their computers, tables, and walls.

There are a number of features that must be improved. Currently, we only support Java-based applications and users cannot directly interchange information between other applications that are not written in Java (such as PowerPoint) or native desktop environments (such as the Windows desktop).

We are also interested in implementing a smaller version of InfoTable for individual users. In this environment, user can hyperdrag items from their computer to the wall (typically a cubicle partition) in front of them, in the same way that they usually attach a post-it note to it. When the user wants to attach a To-Do item on the schedule, he/she can simply hyperdrag it to the physical calendar on the wall.

ACKNOWLEDGMENTS

We thank Takahashi Totsuka for helpful discussions and we are also indebted to Mario Tokoro for their continuing support of our research.

REFERENCES

1. Steven Feiner and A. Shamash. Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers. In *Proceedings of UIST'91, ACM Symposium on User Interface Software and Technology*, pp. 9-17, November 1991.
2. Jorg Geisler. Shuffle, throw or take it! working efficiently with an interactive wall. In *CHI'98 summary*, February 1998.
3. Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards seamless interfaces between people, bits and atoms. In *CHI'97 Proceedings*, pp. 234-241, 1997.
4. Myron W. Krueger. *Artificial Reality II*. Addison-Wesley, 1990.
5. Beth M. Lange, Mark A. Jones, and James L. Meyers. Insight Lab: An immersive team environment linking paper, displays, and data. In *CHI'98 Proceedings*, pp. 550-557, 1998.
6. W.E. Mackay, D.S. Pagani, L. Faber, B. Inwood, P. Louniainen, L. Brenta, and V. Pouzol. Ariel: augmenting paper engineering drawings. In *CHI'95 Conference Companion*, pp. 420-422, 1995.
7. Torsten Holmer Norbert A. Streitz, Jorg Geisler. Roomware for cooperative buildings: Integrated design of architectural spaces and information spaces. In Norbert A. Streitz and Shin'ichi Konomi, editors, *Cooperative Buildings - Integrating Information, Organization, and Architecture*, 1998.
8. Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *SIGGRAPH'98 Proceedings*, pp. 179-188, 1998.
9. Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, pp. 31-39, October 1997.
10. Jun Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Proc. of Asia Pacific Computer Human Interaction (APCHI '98)*, July 1998.
11. Jun Rekimoto. A multiple-device approach for supporting whiteboard-based interactions. In *Proceedings of CHI'98*, February 1998.
12. Stott Robertson, Cathleen Wharton, Catherine Ashworth, and Marita Franzke. Dual device user interface design: PDAs and interactive television. In *CHI'96 Proceedings*, pp. 79-86, 1996.
13. Peter Robinson, Dan Sheppard, Richard Watts, Robert Harding, and Steve Lay. Animated Paper Documents. In *7th International Conference on Human-Computer Interaction, HCI'97*, 1997.
14. Eric Saund. ZombieBoard project description. <http://www.parc.xerox.com/spl/members/saund/zombieboard-public.html>.
15. Itiro Siio. InfoBinder: a pointing device for a virtual desktop system. In *6th International Conference on Human-Computer Interaction (HCI International '95)*, pp. 261-264, July 1995.
16. Questin Stafford-Fraser and Peter Robinson. Brightboard: A video-augmented environment. In *CHI'96 proceedings*, pp. 134-141, 1996.
17. M. Stefik, G. Foster, D. Bobrow, K. Khan, S. Lanning, and L. Suchman. Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Communication of the ACM*, Vol. 30, No. 1, pp. 32-47, 1987.
18. Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. media-Blocks: Physical containers, transports, and controls for online media. In *SIGGRAPH'98 Proceedings*, pp. 379-386, 1998.
19. John Underkoffler. A view from the Luminous Room. *Personal Technologies*, Vol. 1, No. 2., June 1997.
20. John Underkoffler and Hiroshi Ishii. Illuminating Light: An optical design tool with a luminous-tangible interface. In *CHI'98 Proceedings*, pp. 542-549, 1998.
21. Pierre Wellner. The DigitalDesk calculator: Tangible manipulation on a desk top display. In *Proceedings of UIST'91, ACM Symposium on User Interface Software and Technology*, pp. 27-34, November 1991.
22. Pierre Wellner. Interacting with paper on the DigitalDesk. *Communication of the ACM*, Vol. 36, No. 7, pp. 87-96, August 1993.