Vol. 45 No. 10 IPSJ Journal Oct. 2004

Regular Paper

TACT: Mobile Terminal for Session Manipulation

Michimune Kohno,† Yuji Ayatsuka† and Jun Rekimoto†

The goal of wireless networking is to enable service availability anytime and anywhere, but communication between various computing objects is still hard to achieve. If a user can easily establish a connection between the user's terminal and surrounding computers, the user can temporarily utilize the computers' resources, such as a display or a camera. Many projects are tackling this problem, but the user operation issue remains unresolved. This paper describes TACT, a newly created mobile device which can be used as a general interaction tool for ubiquitous computing environments. This device is dedicated to the operation of various kinds of *sessions*; e.g., voice telecommunication and file transfer between remote/surrounding computers. A user with TACT can freely and intuitively manipulate the endpoints of a particular session. As a demonstration of TACT, this paper also describes a system that provides easy-to-use operations to start a videoconference. We demonstrate that the transfer of the addresses of surrounding computers through an established voice session is a useful way to realize interaction in ubiquitous computing environments.

1. Introduction

Wireless networking is aimed at enabling service availability anytime and anywhere, but communication between computing objects is still hard to achieve. We are typically surrounded by a number of computing objects — for example, many people always carry a mobile phone and often have a notebook computer with them. These devices are both wireless networking objects, but connecting them requires a cable and is inconvenient. In addition, the kinds of data they can exchange are strictly limited; for instance, they cannot even synchronize the current time.

A truly ubiquitous computing environment will require that a user be able to temporarily utilize the resources of surrounding embedded computers, including their displays or cameras. To achieve this will require some means allowing the user to easily make a connection between the user's terminal and those computers and operate them. However, it will be troublesome for the user to have to input or choose the network address of the corresponding computer because it will be hard for the user to know such addresses if many computing objects are available. A more workable approach would allow users to remain unaware of underlying network technology details, such as IP addresses.

Our goal is to develop the fundamental technology and an actual device that will enable the above features. Towards this end, we have created TACT as a general interaction tool for ubiquitous computing environments (**Fig. 1**). This is a mobile device that is dedicated to the operation of various kinds of sessions, including audio, video, and data transmission between remote and surrounding computers. We designed TACT as a replacement for current mobile phones so that people can conveniently carry it with them at all times. With TACT, whenever a user wants to utilize a nearby computing resource, the user can open a session and manipulate its endpoints without caring about network addresses.

† Sony Computer Science Laboratories, Inc.

2. Related Work

Our work is essentially based on Rekimoto's "pick-and-drop" [1] technique. "Pick-and-drop" was originally designed as an intuitive user interface to use when transferring a file between different computers. We extended this idea by incorporating a telephone metaphor [2),3)]. Therefore, this paper focuses on the details of integration of "pick-and-drop" into a small mobile device and appliances.

Several research projects have been aimed at achieving effective session management in a ubiquitous computing environment. Harter, et al. proposed BatTeleporting [4] which redirects an X-Window console to the nearest user display detected by the ActiveBat system. Redirection of a window session is implemented by using Virtual Network Computing (VNC). While this provides a simple operation for redirection, more complex operations are difficult because ActiveBat has only two buttons and

Fig. 1 Snapshots of TACT.

no LCD screen. In addition, the destination of a session teleport depends solely on location. Thus, if there are more than two candidates near the user, the Badge button must be repeatedly pressed. In our work, we incorporate alternative, but intuitive user actions to target a candidate resource.

Satchel system[5] is a context-aware system that allows us to easily process our personal documents in an unfamiliar computing environment. This system introduces a token as a unit to access a document, and tokens can be transmitted via an infrared transmitter. Although they are similar to our ideas, the Satchel system does not take account of session control functions between remote locations. In our work, we solve it by using fundamental features of Voice over IP (VoIP).

XWand[6] is an interactive tool for intelligent environments. A user points the wand at a device and control it using simple gestures. Since the system recognizes the three-dimensional position and orientation of the wand for user operations, it is difficult to use the wand at a place where such system is not installed. In addition, gesture operations are not always effective. For example, when a user turns a light on in a crowded room, it is much easier to push a button rather than shaking a stick.

The Aware Home[7] is a set of projects aimed at creating a home environment which is aware of an occupant's whereabouts and activities. It contains various context-aware subprojects. Tran developed a system that helps a resident remember past activities[8]. Since this is a video-based system, the resident does not need to carry anything to be supported. This approach is effective in assisting human memory, but does not otherwise make any operation easier. In other words, the resident can be reminded of or given information by the system, but the overall operation procedure is not changed.

To integrate the controls of audio sessions into a single mobile device, IP-phone is an adequate option. There are many projects that exploit IP-phone (e.g.,[9]~[11]), and most of these have focused on how to utilize context information, such as location and presence. In contrast, in this paper we focus on providing an easy way to open connections to surrounding computers by integrating intuitive user interaction techniques. Other research projects regarding universal remote control (e.g., Pebbles[12]) have mainly focused on how to render various control panels on a small LCD screen, but have not addressed the problem of how the user can initially connect to a device in a user-friendly manner.

For the integration of audio and data sessions, desktop conference tools (e.g., NetMeeting and Polycom) and several instant messenger applications are already available. They provide high-quality IP-based audio/video connections, and can launch another session to share data. However, these tools and applications can share documents only at the computer on which they are running. Our approach is to try to share documents by exploiting surrounding computers.

With regard to the networking aspects, several protocols have been proposed for mobile ad-hoc networks. This paper focuses on software platforms that support context-aware systems. For example, Dey, et al. proposed a context-based software infrastructure solution for smart environments[13], and provided uniform support for building and executing context-aware applications. We have adopted their ideas and UPnP [14] to enable target discovery within a connection link and communication to the targets.

3. Design of the Session Manipulator

In this section, we discuss the features needed in a session manipulation device. We designed TACT by taking into consideration two issues: (1) how to easily establish a connection between computing objects, and (2) the integration of session control functions into a portable communication device like a mobile phone.

3.1 Requirements

Considering the above issues, we decided the basic features of TACT should be the following:

(1) *Portability*: it should be able to replace current mobile phones.

(2) *Close-range communication*: a user should be able to specify the connection device which will work with the user's device without having to know network addresses. A user should be able to operate the computing resources of the connected device and transfer files from/to the device.

(3) *Remote communication*: a user should be able to transfer audio, data files, and presence information to/from a remote user.

(4) *Session control*: a user should be able to generate a session that is subordinate to an already connected session.

A user may want to transfer private data though an established connection. Therefore, a session must be both easy to start and secure to ensure that nobody can snoop among the session packets.

3.2 Implementation Choices

To satisfy these requirements, we decided to apply the following design principles.

To meet the second requirement, we adopted infrared (IR) communication for establishing the initial connection. Communication via IR signals requires that a sender and a receiver be physically located with a line-of-sight channel. Thus, we can consider the two devices as physically collocated when they can exchange their identities via IR signals. By using this condition as a trigger to open a connection, a user no longer needs to input a destination address; the user only has to bring the TACT closer to the target IR receiver.

We adopted several input/output devices for TACT. When it connects to a neighboring computing object, it becomes a dedicated remote control by downloading the menu and event table from the object. What will happen when an input device is activated (e.g., when a TACT switch is pressed) depends on the current event table.

In addition, an object may be a conventional PC. If TACT can become its remote mouse, a user becomes capable of at least basic operations on the PC. Therefore, we mounted a mouse pointer on TACT and defined a packet format to transmit mouse and keyboard events.

TACT can be used as a tool to transfer a file from one device to another. A user can pick up a file and drop it at the destination object without considering their network names (i.e., DNS hostnames or IP addresses). We provide two ways to transfer a file. The first one is the straightforward way; that is, sending the actual contents of a file between TACT and a connected target. The second one is to send/receive only a small fragment that represents the resource information. The actual data is directly transferred to a device when it tries to access the resource. Since it is important to protect privacy [15], we have defined an access ticket that represents the access control information of a resource. The ticket contains the resource location (URL), an expiration date, and the identities of the accessible hosts. It also contains a message integrity check value signed by the owner device of the resource.

As the size of an access ticket is independent of the actual file size, a user can carry many tickets even with a limited amount of memory.

Our approach to meeting the third requirement was to utilize an H.323 VoIP implementation. Since the H.323 protocol defines a packet type that can transport any arbitrary string message, we exploited it by encapsulating various data types into this packet type. We defined an internal message format encapsulated into the string field.

To meet the last requirement, we built a mechanism that allows TACT to exchange an endpoint address of the current close-range connection via an active VoIP session. This allows us to pick up, drop, and create a session endpoint. This mechanism also enables the creation of a subordinate session to an already established VoIP session.

4. Implementation

This section describes the implementation of TACT and a typical appliance that can connect to TACT.

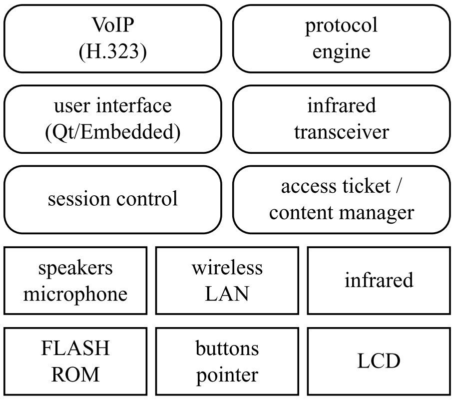


Fig. 2 TACT hardware and software components.

4.1 TACT

According to the implementation choices described in the previous section, we created TACT, a prototype handheld terminal. (Figure 1 shows snapshots of TACT.) The overall software and hardware components are shown in **Fig. 2**.

TACT incorporates a 196-MHz Dragonball MX.1 CPU, a 64-MB SDRAM, and a 32-MB FLASH ROM. A 802.11b wireless LAN is used as a network interface. An IrDA device is mounted for close-range ID transmission. Since this device can also transmit conventional remote control signals, we can also control commercial electronic devices, such as TVs and VCRs.

As input devices, four push buttons, an eight-direction pointer, and a jog dial are mounted. TACT also has a small LCD screen, a pair of speakers, and a microphone.

As an extended feature, we adopted a USB host interface for wired connection. This allows us to connect various kinds of commercial peripheral devices, such as a digital camera or a printer.

We used Linux as the TACT operating system because this allows us to use a number of easily available software resources. We wrote several device drivers for TACT's dedicated hardware devices.

We used Qt/Embedded as the application framework because this allows us to write programs without depending on the underlying hardware platform. To run Qt/Embedded on TACT, we also wrote some hardware adaptor classes.

We wrote the VoIP component by using an OpenH323[16] library. It runs as a single process and communicates with other components through a UNIX domain socket. It makes or hangs up a phone call as requested by the protocol manager. It also listens to an incoming call and notifies the protocol engine of an event when a call arrives. It is responsible for sending and receiving the special packets encapsulated in H.323 string message packets.

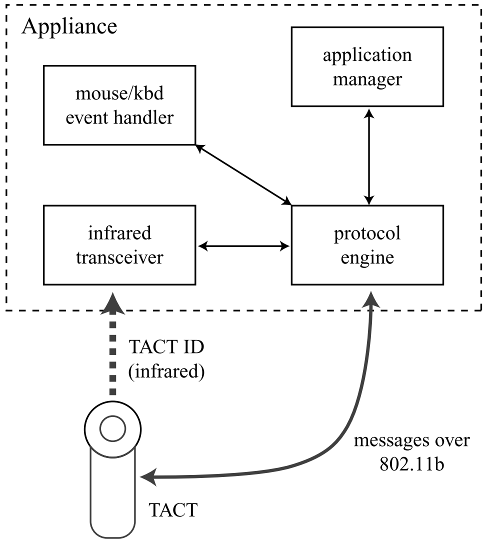


Fig. 3 Components of an appliance.

4.2 Appliance

We have created several prototype appliances by using Windows PCs. The visibility of a PC itself depends on the type of appliance because each has its own external appearance.

An appliance system consists of four components: an event handler, application manager, infrared transceiver, and protocol engine. **Figure 3** illustrates the relationships between these components.

The protocol engine is responsible for communication with TACT. It recognizes the type and body of a packet from TACT and dispatches other components according to the type.

The infrared transceiver component opens an infrared device and waits for a TACT ID. When it detects a signal, it tries to convert the signal to a network address by querying an address server, which is responsible for the mapping table of the TACT ID and network address. The mapping table is automatically maintained since TACT registers itself to the server when it becomes active and when its network configuration has changed.

When the protocol engine receives a mouse/keyboard event message, it requests that the

Fig. 4 Opening a connection with TACT.

mouse/keyboard event handler emulate the corresponding event. The handler converts the request to a Windows message.

The application manager component maintains the session information. When TACT drops an access ticket, the protocol engine requests that it start an appropriate application program. It then launches a new process according to the content type of the ticket. In our current implementation, the filename extension of a resource location is used.

4.3 Example of TACT Use

Figure 4 shows an actual use of TACT. As shown, a user opens a connection between TACT and an appliance (APP) by bringing it closer to the IR receiver of the APP. When a connection is established, TACT downloads a menu and an event table from the APP and renders the menu on its LCD. Since the APP in this case is a picture browser, the user can scroll and magnify pictures by rotating the jog dial.

When the user picks up a picture by pushing a button, the APP generates an access ticket and gives it to TACT. The user can then drop the ticket into another appliance (e.g., a wall projector) in a difference place to show the picture.

Note that while a connection is established, the white LEDs of both TACT and the APP blink synchronously. We implemented this feature to visualize invisible wireless connections.

5. TACT and Videoconferencing

Videoconference systems, such as NetMeeting and Polycom, are often used in office work. When the number of participants is fairly large, a large screen is used to display the peer's scene.

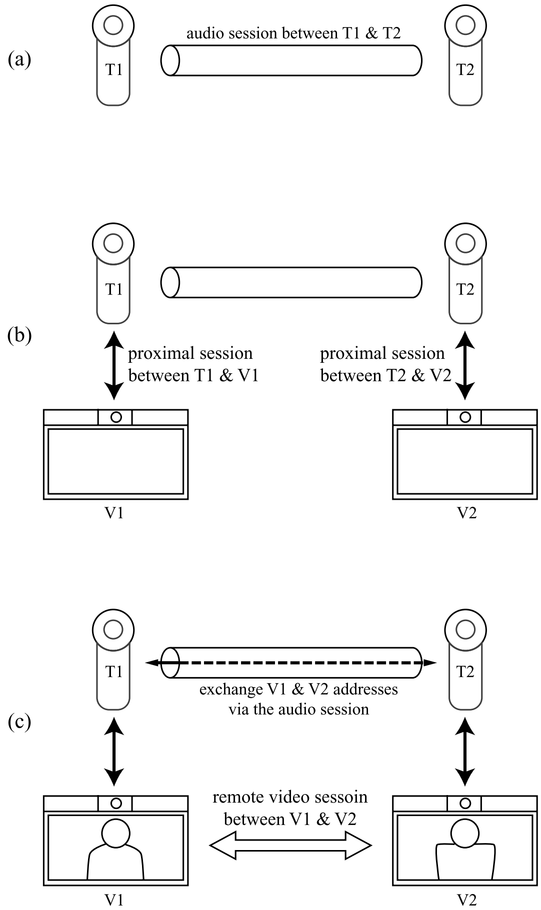


Fig. 5 Connection sequence for starting a videoconference session by using TACT.

Since such a system tends to be placed at a fixed, shared location (e.g., a meeting room), the address book stored in the system might not be useful. Therefore, the most troublesome operation in its usage is initiating a new video session. We have observed many users telling each other over mobile phones what the conference system addresses are. Users often take five minutes or more to set up a session.

This problem is easily solved with TACT. **Figure 5** illustrates the sequence of this task. A user who wants to begin a videoconference first makes a phone call with TACT (a). The user then looks for a conference system having a display and a camera, and opens a connection to the system by physically bringing the terminal closer (b). When the connection is established, the terminal immediately becomes a dedicated remote control of the system and a menu appears on the LCD screen. The menu contains "start videoconference," which trans-

Fig. 6 Snapshots of starting a videoconference.

fers the address of the conference system to the peer TACT with which the user is talking. The address information is encapsulated into an H.323 string message and transmitted to the conference system (c). The systems then establishes a video session.

Therefore the user only has to perform a few simple actions (pushing a jog dial button and a single click for confirmation) to begin a videoconference. **Figure 6** shows snapshots of this scenario.

6. Discussion

6.1 VS. Mobile Phones

As explained in the Introduction, the ability to exploit various kinds of surrounding computing resources is a key feature of ubiquitous computing. Even though some types of current mobile phones can transmit data through IR ports, a user has to choose a deep menu item to use the function, and the number of available operations is limited. In addition, since the entire data is transmitted via IR, a line-of-sight channel between the devices must be maintained for a fairly long time. As a result, the possibility of failure becomes too high.

In contrast, we have focused on how a user can easily operate the surrounding resources. TACT only transmits its ID via IR. Since all other communication is done via wireless LAN, we do not have to maintain a line-of-sight connection for a long time.

6.2 Sessions

We provide two ways to close a session. First, we display the close menu item on the LCD. When a user chooses it, the current connection is disconnected. The second way is more interesting. If a user opens a videoconference by using a VoIP connection (as described in the previous section), closing the VoIP connection automatically closes the videoconference session. Since TACT manages the session information, all subordinate sessions are immediately closed when the master session is closed. This is improvement over previous systems.

From a usability viewpoint, each endpoint of an appliance should provide only a small number of services, just like “stationery.” Otherwise a user has to choose from a menu on a small screen. Alternatively, incorporating the idea proposed by Holland[17] will make the user's choice easier. He proposed Direct Combination (DC) which is a principle for reducing the degree of search, time, and attention required by users to carry out actions.

6.3 Hardware Considerations

At present, the most difficult problem regarding the TACT hardware is the battery life. The current battery life is only 1.5 hours. We are planning to improve this by incorporating a more energy efficient system. Shih's work[18] will be helpful for this.

Mounting a camera on TACT is an interesting improvement. It allows us to take a picture and immediately send it to a friend, a notebook computer, or a nearby printer through simple operations.

Since TACT has only four buttons, though, it is difficult to input text. We are therefore considering development of an attachable keypad.

7. Conclusion

TACT is a mobile terminal designed to be a general interaction tool for ubiquitous computing environments. In this paper, we have explained the concept of a session manipulator and its design principles. We have built and tested preliminary TACT prototypes and some sample appliances.

The videoconference system described in this paper demonstrates the essence of TACT. We have also built other types of interactive system controllable with TACT. With these systems, a user can easily control multiple sessions, acting much like an event "conductor."

We now plan to establish a security model for ubiquitous computing environments. The access ticket model described in this paper will be effective, but it is required to decide what types of information should be included in a ticket.

Acknowledgments We are grateful to Haruo Oba for the appearance design of both TACT and the proximity sensor devices. We are also grateful to the members of the Wearable Solution Division, Sony Corporation, for the hardware design and implementation of TACT. We thank Satoko Nagara for proofreading our manuscript, and express our appreciation to all other members of the Interaction Laboratory, Sony CSL, for their support.

References

1) Rekimoto, J.: Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments, *11th ACM Symposium on User Interface Software and Technology* (*UIST'97*), pp.31–39 (1997).

2) Kohno, M. and Rekimoto, J.: New Generation of IP-Phone Enabled Mobile Devices, *4th International Symposium on Human Computer Interaction with Mobile Devices* (*Mobile-HCI2002*), pp.319–323 (Sep. 2002).

3) Rekimoto, J., Ayatsuka, Y., Kohno, M. and Oba, H.: Proximal Interactions: A Direct Manipulation Technique for Wireless Networking, *INTERACT2003*, pp.511–518 (Sep. 2003).

4) Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P.: The Anatomy of a Context-Aware Application, *Proc. MOBICOM'99*, pp.59–68 (Sep. 1999).

5) Lamming, M., Eldridge, M., Flynn, M., Jones, C. and Pendlebury, D.: Satchel: Providing Access to Any Document, Any Time, Anywhere, *ACM Trans. Computer-Human Interaction*, Vol.7, No.3, pp.322–352 (2000).

6) Wilson, A. and Shafer, S.: XWand: UI for Intelligent Spaces, *Proc. ACM SIGCHI2003*, pp.545–552 (April 2003).

7) Moore, D.J., Essa, I.A. and Hayes, M.H. III: Exploiting Human Actions and Object Context for Recognition Tasks, *Proc.IEEE International Conference on Computer Vision*, pp.80–86 (Sep. 1999).

8) Tran, Q., Truong, K., Mynatt, E. and Abowd, G.: What Was I Cooking?, *Part of the "As We May Live" Demo invited to ACM2001* (March, 2001).

9) Dietz, P.H. and Yerazunis, W.S.: Real-Time Audio Buffering for Telephone Applications, *Proc. 14th ACM Symposium on User Interface Software and Technology* (*UIST'01*), pp.193–194 (2001).

10) Schmandt, C., Kim, J., Lee, K., Valljo, G. and Ackerman, M.: Mediated Voice Communication via Mobile IP, *Proc. 15th ACM Symposium on User Interface Software and Technology* (*UIST'02*), Vol.4, pp.141–150 (Oct. 2002).

11) Hibino, S. and Mockus, A.: handiMessenger: Awareness-Enhanced Universal Communication for Mobile Users, *4th International Symposium on Human Computer Interaction with Mobile Devices* (*MobileHCI2002*), pp.170–183 (Sep. 2002).

12) Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R. and Pignol, M.: Generating Remote Control Interfaces for Complex Appliances, *Proc. 15th ACM Symposium on User Interface Software and Technology* (*UIST'02*), Vol.4, pp.161–170 (Oct. 2002).

13) Dey, A.K., Abowd, G.D. and Salber, D.: A Context-Based Infrastructure for Smart Environment, *Proc. 1st International Workshop on Managing Interactions in Smart Environments* (*MANSE'99*), pp.1–6 (1999).

14) Microsoft Corporation: Understanding Universal Plug and Play: A White Paper. http://www.upnp.org/ (2000).

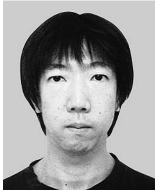
15) Stajano, F.: The Resurrecting Duckling — what next?, *Proc. 8th International Workshop on Security Protocols*, Vol.2133 of LNCS, pp.204–214, Springer-Verlag (2000).

16) OpenH323 Project. H.323 standards. http://www.openh323.org/ (1998).

17) Holland, S., Morse, D.R. and Gedenryd, H.: Direct Combination: A New User Interaction Principle for Mobile and Ubiquitous HCI, *4th International Symposium on Human Computer Interaction with Mobile Devices* (*Mobile-HCI2002*), pp.108–122 (Sep. 2002).

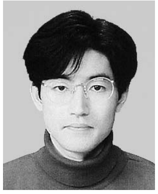
18) Shih, E., Bahl, P. and Sinclair, M.J.: Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices, *Proc. 8th International Conference on Mobile Computing and Networking* (*MOBICOM'02*) (Sep. 2002).

(Received April 10, 2004)
(Accepted July 1, 2004)



Michimune Kohno received his M.E. and Ph.D. degree from Keio University in 1996 and 1999 respectively. He had worked in Sony Computer Science Laboratories, Inc. Since 1999 as an associate researcher he had engaged in research on real-time operating systems. Since 2001 he had been in the Interaction Laboratory in the same company. His current research interests are human interface on ubiquitous computing, computer network and real-time systems.

Yuji Ayatsuka received M.S. degree in Information Science from University of Tokyo in 1995. He joined Sony Corporation in 1998, and joined Interaction Laboratory in Sony Computer Science Laboratories, Inc. in 1999. His research focuses on human computer interaction. He is a member of IPSJ, JSSST, IEEE, and ACM.



Jun Rekimoto received his B.A.Sc., M.Sc., and Ph.D. in Information Science from Tokyo Institute of Technology in 1984, 1986, and 1996, respectively. From 1986 to 1994, he worked for the Software Laboratory of NEC. During 1992–1993, he worked in the Computer Graphics Laboratory at the University of Alberta, Canada, as a visiting scientist. Since 1994 he has worked for Sony Computer Science Laboratories (Sony CSL). In 1999 he formed, and has since directed, the Interaction Laboratory within Sony CSL. At Sony CSL, Rekimoto initiated and has led the "Real-World User Interfaces" project since 1994. This project produced several notable research accomplishments, including NaviCam (a situationally-aware mobile assistant), Pick-and-Drop (a direct-manipulation technique for inter-appliance computing), Multiple-Device Digital Whiteboard, Augmented Surfaces, and TimeScape (a time-machine user interface environment). Some of these are being commercialized in Sony's VAIO personal computer series. Rekimoto's research interests include computer augmented environments, mobile/wearable computing, virtual reality, and information visualization. He has authored dozens of refereed publications in the area of human-computer interactions, including ACM, CHI, and UIST. One of his publications was recognized with the 30th commemorative papers award from the Information Processing Society Japan (IPSJ) in 1992. He also received the Multi-Media Grand Prix Technology Award from the Multi-Media Contents Association Japan in 1998, the Yamashita Memorial Research Award from IPSJ in 1999, and the Japan Inter-Design Award in 2003.