IEEE Intelligent SYSTEMS & their applications

CALL FOR PAPERS

Wearable Computing: Towards Humanistic Intelligence

Submission deadline: Wednesday 14 February 2001
Acceptance decisions: 21 March 2001
Revision deadline: April 11 2001
Publication: May/June 2001 issue

Motivation

Over the past 20 years Wearable Computing has emerged as the perfect tool for embodying Humanistic Intelligence (HI). HI is defined as intelligence that arises from the human being in the feedback loop of a computational process in which the human and computer are inextricably intertwined. When a wearable computer functions in a successful embodiment of HI, the computer uses the human's mind and body as one of its peripherals, just as the human uses the computer as a peripheral. This reciprocal relationship, where each uses the other in its feedback loop, is at the heart of HI.

There are three fundamental operational modes of an embodiment of HI: Constancy, Augmentation, and Mediation. Firstly, there is a constantly of user interface, which implies an "always ready" interactional constancy, supplied by a continuously running operational constancy. Wearable computers are unique in their ability to provide this "always ready" condition which might, for example, include a retroactive video capture for a face recognizing reminder system. After-the-fact devices like traditional cameras and palmtop organizers cannot provide this retroactive computing capability. Secondly, there is an augmentational aspect in which computing is NOT the primary task. Again, wearable computing is unique in its ability to be augmentational without being distracting to a primary task like navigating through a corridor, or trying to walk down stairs. Thirdly, there is a mediational aspect in which the computational host can protect the human host from information overload, by deliberately diminished reality, such as by visually filtering out advertising signage and billboards.

Implicit in the Augmenting and Mediating modes is a spatiotemporal contextual awareness from sensors (wearable cameras, microphones, etc.).

As an example of H.I., it is now possible to build a miniature nearly invisible apparatus for lifelong video capture, that can also predict or infer and distinguish from among threat or opportunity based on previously captured material. Such computing blurs the line between remembering and recording, as well as the line between thinking and computing. Thus we will need a whole new way of studying these new human-based intelligent systems. Such an apparatus has in fact already raised various interesting privacy and accountability issues. Thus HI necessarily raises a whole new set of humanistic issues not previously encountered.

For this special issue we seek papers describing intelligent systems that include the human as an integral part of the system. Preference will be given to papers describing systems that actually demonstrate the integration of human-computer adaptation, intelligent real-time action, reasoning, learning, and control, or that focus on a specific clearly stated problem or clearly stated scientific hypothesis.

Submission Guidelines

Authors should note that IEEE Intelligent Systems is a scholarly peer-reviewed publication that is intended for a broad research and user community. Therefore an informal, direct and lively writing style should be adopted, while at the same time still maintaining a high degree of quality in the actual research that is reported. Manuscripts should be original and should have between 6 and 10 published pages (not more than 7500 words) with up to 10 references. For additional details, please refer to our author guidelines. Manuscripts should be sent to mann@eecg.toronto.edu in uuencoded gzipped PostScript format, along with LaTeX source if present, or including raw ASCII article text (not more than 80 characters per line), if typeset in a program other than LaTeX. If filesize is large, Manuscripts should be passed by reference not by value (e.g. email a URL where a gzipped PostScript file and ASCII text can be retrieved). Proprietary file formats such as msword will not be accepted.

Guest Editor:

Steve Mann
University of Toronto
Department of Electrical and Computer Engineering, Room S.F. 2001,
10 King's College Road; Toronto, Ontario, Canada; M5S 3G4
Tel. 416.946-3387
Fax. 416.971-2326
mann@eecg.toronto.edu

Wearable Computing: Toward Humanistic Intelligence

Steve Mann, *University of Toronto*

Over the past 20 years, wearable computing has emerged as the perfect tool for embodying *humanistic intelligence*. HI is intelligence that arises when a human is part of the feedback loop of a computational process in which the human and computer are inextricably intertwined.

It is common in the field of human–computer interaction to think of the human and computer as separate entities. (Indeed, the term "HCI" emphasizes this separateness by treating the human and computer as different entities that interact.) However, in HI theory, we prefer not to think of the wearer and the computer with its associated I/O apparatus as separate entities. Instead, we regard the computer as a second brain and its sensory modalities as additional senses, which synthetic synesthesia merges with the wearer's senses.

When a wearable computer functions in a successful embodiment of HI, the computer uses the human's mind and body as one of its peripherals, just as the human uses the computer as a peripheral. This reciprocal relationship is at the heart of HI.

Assisting human intelligence

HI also suggests a new goal for signal-processing hardware—that is, in a truly personal way, to directly assist, rather than replace or emulate, human intelligence. To facilitate this vision, we need a simple and truly personal computational signal-processing framework that empowers the human intellect.

The HI framework, which arose in Canada in the 1970s and early 1980s, is in many ways similar to Douglas Engelbart's vision that arose in the 1940s while he was a radar engineer. Engelbart, while seeing images on a radar screen, realized that the cathode ray screen could also display letters of the alphabet and computer-generated pictures and graphical content. Thus, computing could be an interactive experience for manipulating words and pictures. Engelbart envisioned the mainframe computer as a tool for augmented intelligence and communication, which many people in a large amphitheater could use to interact.[1,2]

Although Engelbart did not foresee the personal computer's significance, modern personal computing certainly embodies his ideas. This special issue presents a variety of attempts at realizing a similar vision, but with the computing resituated in the context of the user's personal space. The idea is to move the tools of augmented intelligence and communication directly onto the body. This will give rise not only to a new genre of truly personal computing but also to some new capabilities and *affordances* arising from direct physical proximity to the human body, allowing the HI feedback loop to develop. (Affordances are what an environment offers to an organism.[3]) Moreover, a new family of applications will arise, in which the body-worn apparatus augments and mediates the human senses.

HI theory

HI's goals are to work in extremely close synergy with the human user and, more important, to arise partly because of the very existence of the human user.[4] HI achieves this synergy through a user interface to signal-processing hardware that is in close physical proximity to the user and is continuously accessible.

1094-7167/01/$10.00 © 2001 IEEE

Operational modes

An embodiment of HI has three fundamental operational modes: *constancy*, *augmentation*, and *mediation*.

Constancy. An embodiment of HI is *operationally constant*; that is, although it might have power-saving (sleep) modes, it is never completely shut down (as is typically a calculator worn in a shirt pocket but turned off most of the time). More important, it is also *interactionally constant*—that is, the device's inputs and outputs are always potentially active. Interactionally constant implies operationally constant, but operationally constant does not necessarily imply interactionally constant.

So, for example, a pocket calculator kept in your pocket but left on all the time is still not interactionally constant, because you cannot use it in this state (you still have to pull it out of your pocket to see the display or enter numbers). A wristwatch is a borderline case. Although it operates constantly to keep proper time and is conveniently worn on the body, you must make a conscious effort to orient it within your field of vision to interact with it.

Wearable computers are unique in their ability to provide this always-ready condition, which might, for example, include retroactive video capture for a face-recognizing reminder system. After-the-fact devices such as traditional cameras and palmtop organizers cannot provide such retroactive computing.

Figure 1a depicts the signal flow from human to computer, and computer to human, for the constancy mode.

Once, people did not see why devices should be operationally and interactionally constant; this shortsighted view led to the development of many handheld or so-called "portable" devices. In this special issue, however, we will see why it is desirable to have certain personal-electronics devices, such as cameras and signal-processing hardware, always on—for example, to facilitate new forms of intelligence that assist the user in new ways.

Augmentation. Traditional computing paradigms rest on the notion that computing is the primary task. Intelligent systems embodying HI, however, rest on the notion that computing is not the primary task. HI assumes that the user will be doing something else while computing, such as navigating through a corridor or walking down stairs. So, the computer should augment the intellect or the senses, without distracting a primary task. Implicit in this mode is a spatiotemporal contextual awareness from sensors (wearable cameras, microphones, and so on).

Figure 1b depicts the signal flow between the human and computer in this mode.

Mediation. Unlike handheld devices, laptop computers, and PDAs, good embodiments of HI can *encapsulate* the user (see Figure 1c). Such an apparatus doesn't necessarily need to completely enclose us. However, the basic concept of mediation allows for whatever degree of encapsulation is desired (within the limits of the apparatus), because it affords us the possibility of a greater degree of encapsulation than traditional portable computers. As with the augmentation mode, a spatiotemporal contextual awareness from sensors is implicit in this mode.

The encapsulation that mediation provides has two aspects, one or both of which can be implemented in varying degrees, as desired.

The first aspect is *solitude*. The ability to mediate our perception lets an embodiment of HI act as an information filter. For example, we can block out material we might not wish to experience (such as offensive advertising) or replace existing media with different media (for example, see the "Filtering Out Unwanted Information" sidebar). In less extreme manifestations, it might simply let us moderately alter aspects of our perception of reality. Moreover, it could let us amplify or enhance desired inputs. This control over the input space contributes considerably to the most fundamental HI issue: user empowerment.

The second aspect is *privacy*. Mediation lets us block or modify information leaving our encapsulated space. In the same way that ordinary clothing prevents others from seeing our naked bodies, an embodiment of HI might, for example, serve as an intermediary for interacting with untrusted systems, such as third-party implementations of digital anonymous cash. In the same way that martial artists, especially stick fighters, wear a long black robe or skirt that reaches the ground to hide the placement of their feet from their opponent, a good embodiment of HI can clothe our otherwise transparent movements in cyberspace and the real world.

Other technologies such as desktop computers can, to a limited degree, help us protect our privacy with programs such as Pretty Good Privacy. However, the primary weakness of these systems is the space between them and their user. Compromising the link between the human and the computer (perhaps through a Trojan horse or other planted virus) is generally far easier when they are separate entities.

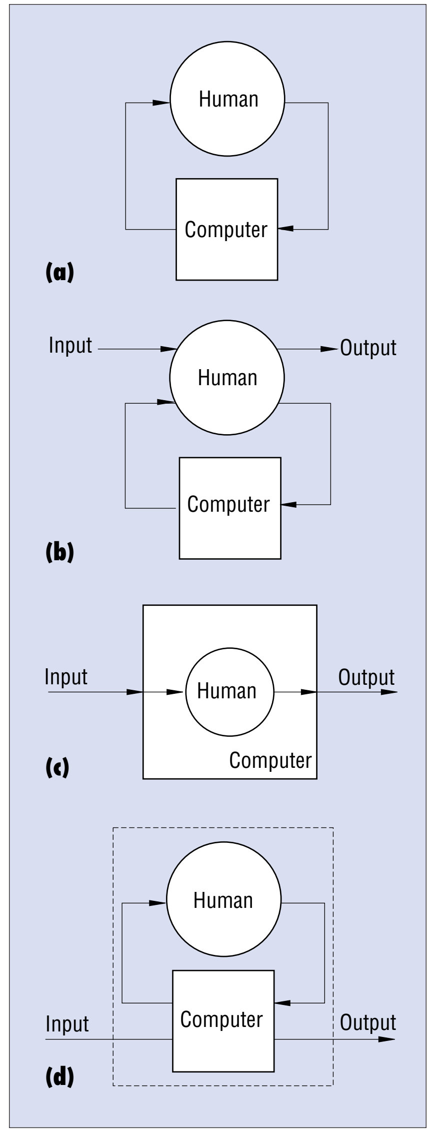


Figure 1. Signal flow paths for the three basic operational modes of devices that embody HI: (a) constancy; (b) augmentation; (c) mediation; (d) mediation (redrawn to resemble Figures 1a and 1b) emphasizing the separate protective shell that encapsulation can provide.

A personal information system that the wearer owns, operates, and controls can provide a much greater level of personal privacy. For example, if the user always wears it (except perhaps during showering), the hardware is less likely to fall prey to attacks. Moreover, the close synergy between the human and computer makes the system less vulnerable to direct attacks, such as someone

Filtering Out Unwanted Information

The owner of a building or other real estate can benefit financially from placing advertising signs in the line of sight of all who pass by the property (see Figure A1). These signs can be distracting and unpleasant. Such theft of solitude benefits the owner at the expense of the passersby.

Legislation is one possible solution to this problem. Instead, I propose a *diffusionist*[1] approach in the form of a simple engineering solution that lets the individual filter out unwanted real-world spam. Such a wearable computer, when functioning as a reality mediator, can create a modified perception of visual reality (see the coordinate-transformed images in Figure A2). So, it can function as a visual filter to filter out the advertising in Figure A1 and replace it with useful subject matter, as in Figure A3. Such a computer-mediated intelligent-signal-processing system is an example application of humanistic intelligence.

Reference

1. S. Mann, "Reflectionism and Diffusionism," *Leonardo*, vol. 31, no. 2, 1998, pp. 93–102; http://wearcam.org/leonardo/index.htm (current 5 June 2001).

Figure A. Filtering out unwanted advertising messages (each row shows frames from a movie): (1) Advertising can be distracting and annoying. (2) A wearable computing device together with an EyeTap system (see the other sidebar) creates a modified perception of the advertising. (3) It then replaces the advertising with subject matter useful to the user.

looking over your shoulder while you're typing or hiding a video camera in the ceiling above your keyboard.

For the purposes of this special issue, we define privacy not so much as the absolute blocking or concealment of personal information, but as the ability to control or modulate this outbound information channel. So, for example, you might wish members of your immediate family to have greater access to personal information than the general public does. Such a family-area network might feature an appropriate access control list and a cryptographic communications protocol.

In addition, because an embodiment of HI can encapsulate us—for example, as clothing directly touching our skin—it might be able to measure various physiological quantities.

Thus, the encapsulation shown in Figure 1c enhances the signal flow in Figure 1a. Figure 1d makes this enhanced signal flow more explicit. It depicts the computer and human as two separate entities within an optional protective shell, which the user can fully or partially open if he or she desires a mixture of augmented and mediated interaction.

Combining modes. The three modes are not necessarily mutually exclusive; constancy is embodied in augmentation and mediation.

These last two are also not necessarily meant to be implemented in isolation. Actual embodiments of HI typically incorporate aspects of augmentation and mediation. So, HI is a framework for enabling and combining various aspects of each of these modes.

Basic signal flow paths

Figure 2 depicts the six basic signal flow paths for intelligent systems embodying HI. The paths typically comprise vector quantities. So, the figure depicts each basic path as multiple parallel paths to remind you of the vector nature of the signals.

Each path defines an HI attribute:

1. *Unmonopolizing.* The device does not necessarily cut you off from the outside world as a virtual reality game or the like does.
2. *Unrestrictive.* You can do other things while using the device—for example, you can input text while jogging or running down stairs.
3. *Observable.* The device can get your attention continuously if you want it to. The output medium is constantly perceptible. It is sufficient that the device is almost always observable, within reasonable limitations—for example, as when a camera viewfinder or computer screen is not visible when you blink your eye.
4. *Controllable.* The device is responsive. You can take control of it at any time. Even in automated processes, you should be able to manually override the automation to break open the control loop and become part of the loop. Examples of this controllability might include a Halt button you can invoke when an application mindlessly opens all 50 documents that were highlighted when you accidentally pressed Enter.
5. *Attentive.* The device is environmentally aware, multimodal, and multisensory. This ultimately gives you increased situational awareness.
6. *Communicative.* You can use the device as a communications medium when you wish. It lets you communicate directly to others or helps you produce expressive or communicative media.

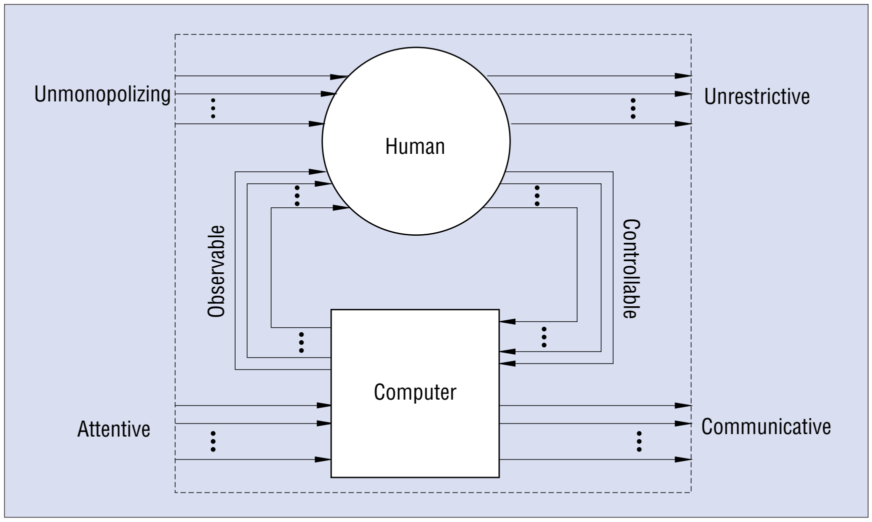


Figure 2. The six signal flow paths for intelligent systems embodying HI. Each path defines an HI attribute.

Adapting to HI

Because devices embodying HI often require that the user learn a new skill set, adapting to them is not necessarily easy. Just as a young child takes many years to become proficient at using his or her hands, some devices that implement HI have taken years of use before they begin to behave like natural extensions of the mind and body. So, in terms of human–computer interaction,[5] the goal is not just to construct a device that can model (and learn from) the user, but, more important, to construct a device from which the user also must learn. Therefore, to facilitate the latter, devices embodying HI should provide a constant user interface that is not so sophisticated and intelligent that it confuses the user. Although the device might implement sophisticated signal-processing algorithms, the cause-and-effect relationship of the input (typically from the environment or the user's actions) to this processing should be clearly and continuously visible to the user.

Accordingly, the most successful examples of HI afford the user a very tight feedback loop of system observability. A simple example is the viewfinder of an EyeTap imaging system (see the related sidebar). In effect, this viewfinder continuously endows the eye with framing, a photographic point of view, and an intimate awareness of the visual effects of the eye's own image-processing capabilities.

A more sophisticated example of HI is a biofeedback-controlled EyeTap system, in which the biofeedback process happens continuously, whether or not the system is taking a picture. Over a long period of time, the user will become one with the machine, constantly adapting to the machine intelligence, even if he or she only occasionally deliberately uses the machine.

This special issue

In their profound and visionary article, Joshua Anhalt and his colleagues provide a background for context-aware computing, along with some practical examples of HI implemented in such forms as a portable help desk. This work comes from Carnegie Mellon University's Software Engineering Institute and IBM's T.J. Watson Research Center. The SEI is under the direction of Daniel Siewiorek, who has been working on wearable computing for many years.

This article marks an interesting departure from their previous work in military equipment maintenance applications, and suggests a branching out into applications more suitable for mainstream culture. Wearable computing has gone beyond the military-industrial complex; we are at a pivotal era where it will emerge to affect our daily lives.

Recognizing the importance of privacy and solitude issues, the authors formulate the notion of a *distraction matrix* to characterize human attentional resource allocation.

Li-Te Cheng and John Robinson also look at an application targeted for mainstream consumer culture. They report on context awareness through visual focus, emphasizing recognition of visual body cues, from the first-person perspective of a personal imaging system. They provide two concrete examples: a memory system for playing the piano and a system for assisting ballroom dancing. This work shows us further examples of how wearable computers have become powerful enough to perform vision-based intelligent signal processing.

EyeTap

One application of humanistic intelligence is an EyeTap.[1] An EyeTap is a nearly invisible miniature apparatus that causes the human eye to behave as if it were both a camera and a display. This device can facilitate lifelong video capture and can determine the presence of an opportunity or a threat, based on previously captured material.

One practical application of an EyeTap is in assisting the visually impaired. In the same way that a hearing aid contains a microphone and speaker with signal processing in between, the EyeTap causes the eye itself to, in effect, contain an image sensor and light synthesizer, with processing in between the two.

The EyeTap tracks depth by using a single control input to manually or automatically focus a camera and an *aremac* together.[1] The aremac ("camera" spelled backwards) is a device that resynthesizes light that was absorbed and quantified by the camera. Figure B diagrams three approaches to depth tracking. Solid lines denote real light from the subject matter, and dashed lines denote virtual light synthesized by the aremac.

Figure B1 shows an autofocus camera controlling the aremac's focus. When the camera focuses to infinity, the aremac focuses so that it presents subject matter that appears as if it is infinitely far. When the camera focuses closely, the aremac presents subject matter that appears to be at the same close distance. A zoom input controls both the camera and aremac to negate any image magnification and thus maintain the EyeTap condition. W denotes rays of light defining the widest field of view. T (for tele) denotes rays of light defining the narrowest field of view. The camera and aremac fields of view correspond.

Figure B2 shows eye focus controlling both the camera and aremac. An eye focus measurer (via the *eye focus diverter*, a beamsplitter) estimates the eye's approximate focal distance. Both the camera and aremac then focus to approximately this same distance.

The mathematical-coordinate trans-

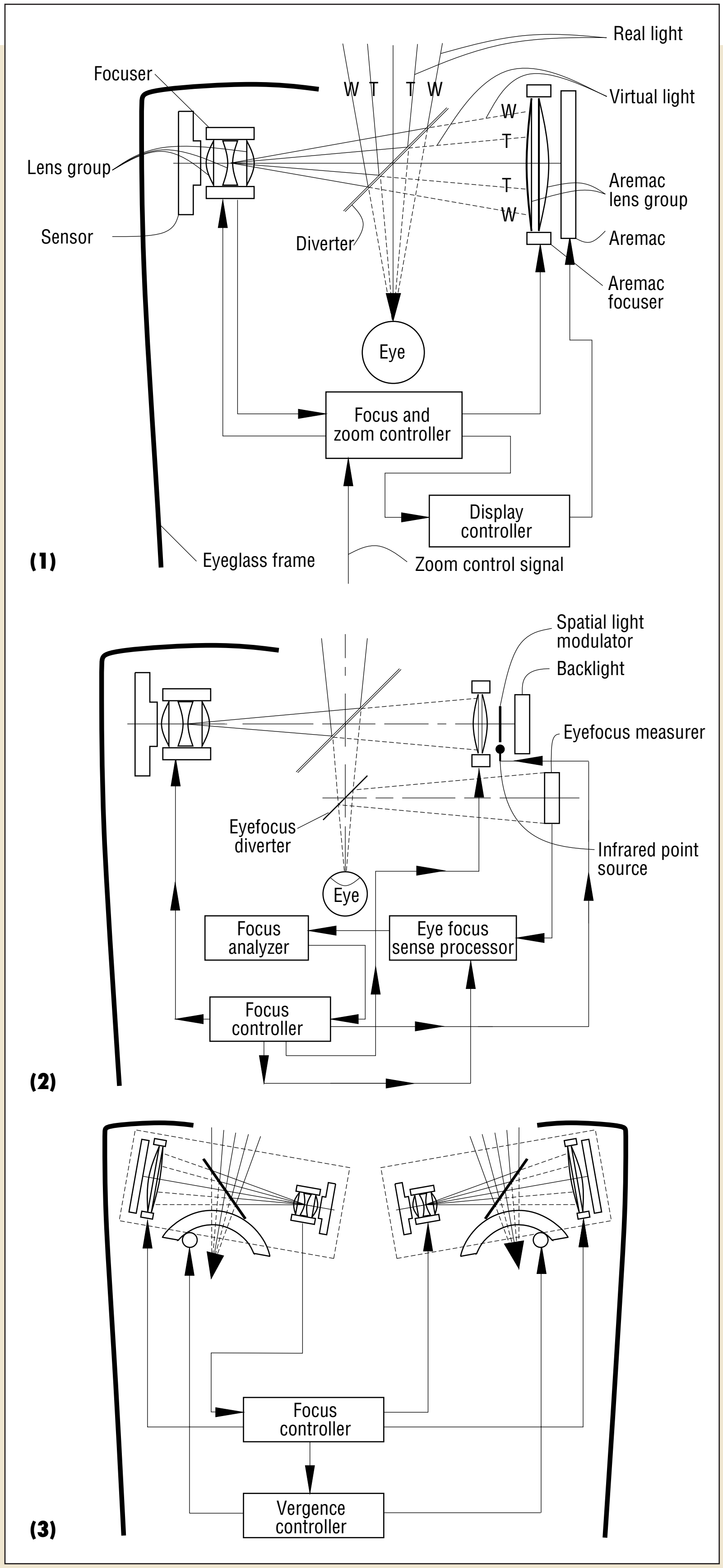


Figure B. Depth tracking with the EyeTap: (a) An autofocus camera controls focus of the *aremac*, which resynthesizes light that was absorbed and quantified by the camera. Solid lines denote real light from the subject matter; dashed lines denote virtual light synthesized by the aremac. W denotes rays of light defining the widest field of view. T (for tele) denotes rays of light defining the narrowest field of view. (b) Eye focus controls both the camera and the aremac. (c) An autofocus camera on the left controls focus of the right camera and both aremacs (as well as vergence).

formations in Figure B2 arise from the system's awareness of the wearer's gaze pattern, such that this intelligent system is *activity driven*. Areas of interest in the scene will attract the human operator's attention, so that he or she will spend more time looking at those areas. In this way, those parts of the scene of greatest interest will be observed with the greatest variety of quantization steps (for example, with the richest collection of differently quantized measurements). So, the EyeTap will automatically emphasize these parts in its composite representation.[1]

This natural *foveation* process arises, not because the EyeTap itself has figured out what is important, but simply because it is using the operator's brain as its guide to visual saliency. Because operating the EyeTap does not require any conscious thought or effort, it resides on the human host without presenting any burden. However, it still benefits greatly from this form of humanistic intelligence.

In Figure B3, an autofocus camera on the left controls the focus of the right camera and both aremacs (as well as the vergence). In a two-eye system, both cameras and both aremacs should focus to the same distance. So, one camera is a focus master, and the other is a focus slave. Alternatively, a focus combiner can average the focus distance of both cameras and then make the two cameras focus at an equal distance. The two aremacs and the vergence controllers for both eyes track this same depth plane as defined by the camera autofocus.

Computing such as the EyeTap provides blurs the line between remembering and recording, as well as the line between thinking and computing. So, we will need a whole new way of studying these new human-based intelligent systems. Such an apparatus has already raised various interesting privacy and accountability issues. Thus, HI necessarily raises a set of humanistic issues not previously encountered in the intelligent systems field.

Reference

1. S. Mann, "Humanistic Intelligence/Humanistic Computing: 'Wearcomp' as a New Framework for Intelligent Signal Processing," *Proc. IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2123–2151; http://wearcam.org/procieee.htm (current 5 June 2001).

Kaoru Sumi and Toyoaki Nishida put context awareness in a spatiotemporal global framework, with computer-based human communication. In the context of conversation, the system illustrates how HI can serve as a human–human communications medium, mediated by wearable computer systems.

David Ross provides an application of HI for assistive technology. Besides the military-industrial complex, early HI adopters might well be those with a visual or other impairment. For this sector of the population, wearable computing can make a major difference in their lives.

Ömer Faruk Özer, Oguz Özün, C. Öncel Tüzel, Volkan Atalay, and A. Enis Çetin describe a personal-imaging system (wearable camera system) for character recognition. Chain-coded character representations in a finite-state machine are determined by way of personal imaging as a user interface.

Soichiro Matsushita describes a wireless sensing headset. Indeed, it has often been said that a good embodiment of HI will replace all the devices we normally carry with us, such as pagers, PDAs, and, of course, cellular telephones. Thus, a context-awareness-enhancing headset is a good example of how HI will improve our daily lives.

Although I have formulated a theoretical framework for humanistic intelligence, the examples I've described in this introduction are not merely hypothetical; they have been reduced to practice. Having formulated these ideas some 30 years ago, I have been inventing, designing, building, and wearing computers with personal-imaging capability for more than 20 years. Actual experience of this sort has grounded my insights in this theory in a strong ecological foundation, tied directly to everyday life.

We are at a pivotal era in which the convergence of measurement, communications, and computation, in the intersecting domains of wireless communications, mobile computing, and personal imaging, will give rise to a simple device we wear that replaces all the separate informatic items we normally carry.

Although I might well be (apart from not more than a dozen or so of my students) the only person to be continuously connected to, and living in, a computer-mediated reality, devices such as EyeTaps and wearable computers doubtlessly will enjoy widespread use in the near future.

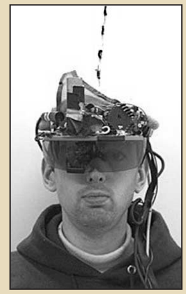
Twenty years ago, people laughed at this idea. Now I simply think of Alexander Graham Bell's prediction that the day would come when there would be a telephone in every major city of this country.

Thus, there is perhaps no better time to introduce HI by way of a collection of articles showing how these ideas can be actually reduced to practice. ■

References

1. D.C. Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, research report AFOSR-3223, Stanford Research Inst., Menlo Park, Calif., 1962; www.histech.rwth-aachen.de/www/quellen/engelbart/ahi62index.html (current 5 June 2001).
2. D.C. Engelbart, "A Conceptual Framework for the Augmentation of Man's Intellect," *Vistas in Information Handling*, P.D. Howerton and D.C. Weeks, eds., Spartan Books, Washington, D.C., 1963, pp. 1–29.
3. J. Zhang, "Categorization of Affordances," Dept. of Health Informatics, Univ. of Texas at Houston, http://acad88.sahs.uth.tmc.edu/courses/hi6301/affordance.html (current 3 July 2001).
4. S. Mann, "Humanistic Intelligence/Humanistic Computing: 'Wearcomp' as a New Framework for Intelligent Signal Processing," *Proc. IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2123–2151; http://wearcam.org/procieee.htm (current 5 June 2001).
5. W.A.S. Buxton and R.M. Baecker, *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, Morgan Kaufmann, San Francisco, 1987, chapters 1 and 2.
6. M. Weiser, *Ubiquitous Computing*, http://sandbox.parc.xerox.com/ubicomp (current 5 June 2001).
7. J. Cooperstock, *Reactive Room*, www.dgp.toronto.edu/~rroom/research/papers (current 5 June 2001).

The Author



Steve Mann is a faculty member at the University of Toronto's Department of Electrical and Computer Engineering. He built the world's first covert fully functional wearable image processor with computer display and camera concealed in ordinary eyeglasses and was the first person to put his day-to-day life on the Web as a sequence of images. He received his PhD in personal imaging from MIT. Contact him at the Dept. of Electrical and Computer Eng., Univ. of Toronto, 10 King's College Rd., S.F. 2001, Canada, M5S 3G4. He can be reached via e-mail at mann@eecg.toronto.edu or by tapping into his right eye, http://eyetap.org.

Personal Contextual Awareness through Visual Focus

Li-Te Cheng *Lotus Research*

John Robinson *University of York*

The authors explore options for using body parts as focal points for wearable systems. Their two working systems, Handel and Footprint, demonstrate several possibilities for recognizing visual body cues.

One of the goals of wearable computer technology is to let users move freely in their environments while interacting with the virtual information that their wearable computers associate with the real-world objects around them.[1] Augmentation and mediation technology can help wearables achieve this goal by gathering and acting on sensor readings of the user's activity and environment. This technology, in other words, can help create an awareness of the user's personal context, which is an increasingly important feature for wearable computer interfaces.[2]

However, programming a wearable computer to sense when it is appropriate to enable interaction is a significant challenge that involves analysis of the user's overall environment. Sensing context presents serious complications, especially because a user's context is never stable. However, even in the complete absence of environmental stability, there still is one physical object on which wearable technology can focus: the user's own body.

One of our ideas is to focus, in particular, on the user's hands and feet. We can apply this technique to wearables in several contexts: physical rehabilitation, choreography, pathfinding, sports, and so forth. By focusing wearable technology on hands and feet, we can define virtual annotations and commands by hand or foot gestures. For example, framing a shot for video or photography could be triggered by a two-handed framing gesture, where the size and location of the framing gesture defines the parameters of the snapshot and the placement of a virtual annotation window.[1] The wide range of possible applications of such technology present new opportunities for mobile computing devices.

To demonstrate how this notion of personal context can enhance specific functions in wearable computers, we built two working systems: Handel and Footprint. We designed both to enhance user learning experiences seamlessly by linking instructional overlays to hands and feet.

Personal context

We define *personal context* as the contextual awareness of the user's body as a stimulus and rendering surface for augmentation and mediation technology. While general context is derived from the environment at large, personal context is derived from an awareness of a user's body parts with respect to a particular task. Performing particular tasks requires that we move our hands and feet in certain routine ways, which suggests a good focal point for any virtual information that a wearable computer might present. For example, if a wearable were to use direct sensor measurements or a combination of sensors and pattern recognition, it could derive personal context from the user's body movements. Virtual information could then augment the user's experience through a heads-up display or through audio feedback. The ultimate goal would be to provide such feedback mediated by how relevant that feedback would be to the task at hand.

Augmented reality systems[3] are designed to overlay virtual information onto the real world. Such systems include first-person applications that use head-mounted displays and environmental sensor cues to overlay information onto appropriate objects to help direct a user in a particular task (such as servicing a printer[4] or reading an enhanced book[5]). A personal-context approach to a printer-servicing application or an augmented book would rely on the user's gaze

1094-7167/01/$10.00 © 2001 IEEE

with respect to the hands rather than on some kind of ultrasonic tracking infrastructure[4] or on specially marked book pages.[5]

Having a wearable computer rely on the user's body as a rendering surface does not necessarily imply a body-stabilized interface such as a cylindrical or spherical overlay that surrounds the user.[6] An object-centric interface—where the objects are really parts of the user's body—appears to the user to be world-stabilized. Unlike a true world-stabilized interface,[6] an object-centric interface attaches information to body parts with little or no attempt to assess a complete world model. In these systems, overlay graphics are linked to the user's hands, and tracking is done using simple two-dimensional techniques that don't require knowledge of a user's physical location. A simplified model based on personal context makes tracking algorithms more readily available because they are somewhat easier to implement than complete world models.

Handel and Footprint both use personal context (as we've defined it here) to infer a user's need for augmentation.

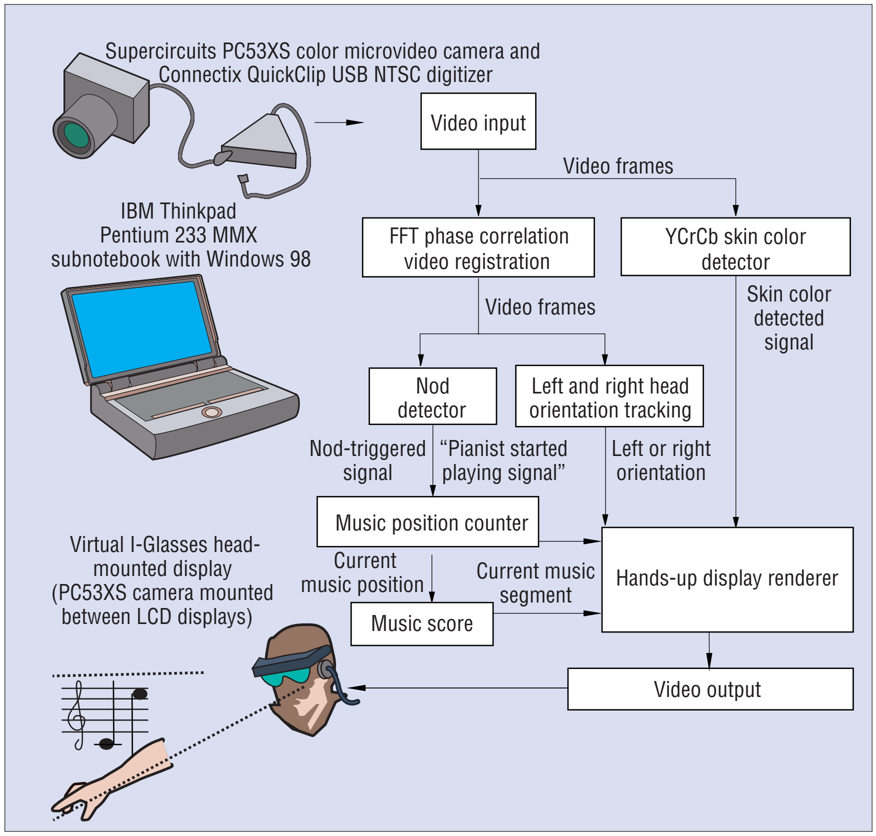


Figure 1. Handel's architecture and components.

Handel: Giving the user a hand

A considerable amount of research exists in the areas of hand-based user interfaces and computer-vision techniques used to locate and recognize hand gestures.[7] Data gloves, magnetic trackers, and optical sensors can all be used to obtain hand orientation. In these cases, however, the hand acts solely as an input device. We designed Handel (*hand*-based *e*nhancement for *l*earning) to rely on hand movements to trigger an augmented-reality overlay onto the user's hands during piano practice. Essentially, Handel creates a "hands-up" display instead of a heads-up display.

There are some preexisting technologies that are similar to Handel. Some of these technologies merge interactive graphics with hands[8] and some even place small displays on hands to ease interaction with large-screen virtual environments.[9] There are also countless piano-teaching tools, including self-help computer software that shows keyboard layouts to guide pianists. Modern acoustic player pianos such as the Disklavier allow direct playback on the keyboard from music files or from captured piano-key action.

In Handel, the pianist is equipped with a wearable computer system and sits at a normal acoustic piano with no sheet music. As the pianist attempts to play a piece from memory, the pianist can look down at the hands to invoke the visual aid. Focusing on the hands is the trigger. If the pianist does not look down, no graphics clutter the screen so that the pianist can concentrate on playing from memory. When the pianist looks at the right hand, Handel shows only the right hand's part of the music at the current position in the piece. Similarly, if the pianist looks directly at the left hand, Handel shows only the music for that hand. Handel uses each hand as an input to trigger the overlay of virtual sheet music. Because Handel presents the music near the relevant hand, the hand also acts as a context-sensitive display window for the sheet music.

Handel uses a head-mounted video camera to perform scene analysis. The pianist's hands are totally unencumbered and free to interact normally with the piano. Handel uses Fast Fourier Transform (FFT) phase-correlation analysis[10] on consecutive video frames to determine whether the pianist's head is looking to the left or to the right. Handel also uses a skin-color table to detect whether a hand is in view or not. Handel sets the skin-color scheme during a training session beforehand. Skin-color detection is sufficient for determining where a pianist is looking, because Handel assumes that the only thing the head-mounted camera will see is the piano. Figure 1 illustrates Handel's general system architecture. On our Pentium 233-MHz subnotebook, Handel runs at about five frames per second.

The practice session begins with the pianist loading the music score into Handel. For the current implementation, we created a simple score language to store the music in a text file. The pianist dons the head-mounted display and sits in front of the piano. He or she then gives a nod when starting to play the memorized music. Handel uses FFT phase correlation to detect a strong vertical displacement (the nod) to begin incrementing an internal counter to keep track of the current position in the piece. In the current implementation, Handel increments the counter at a predetermined rate.

While the pianist plays the piece, Handel doesn't overlay anything on the pianist's heads-up display (Figure 2a) until it sees skin color. When Handel detects skin, it assumes that the pianist is looking down at the hands. Handel determines what hand to overlay on the

Figure 2. Views from the head-mounted display: (a) nothing overlaid when no hand is in view, (b) left-hand part displayed for the left hand, and (c) right-hand part displayed for the right hand.

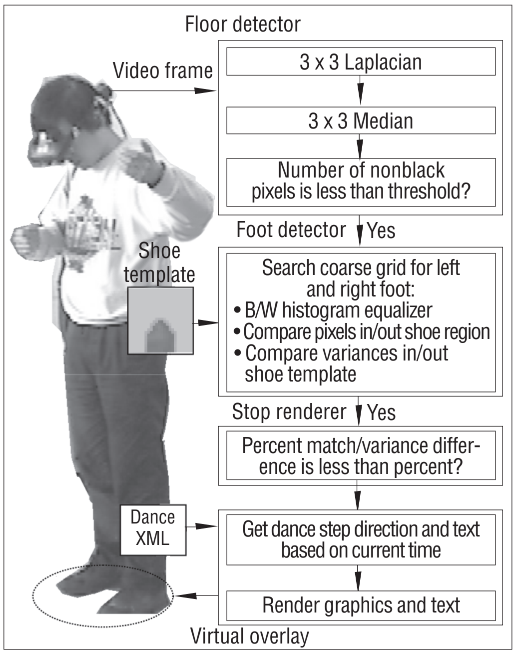


Figure 3. Footprint's system architecture.

basis of whether the pianist is looking to the left (Figure 2b) or to the right (Figure 2c). Handel then displays the musical score at the current position, for the given hand, on the head-mounted display. Meanwhile, the software continues to update itself while the pianist is playing. Handel renders the score at a fixed position on the left side of the display for the left hand and on the right for the right hand. Handel doesn't link the score to the hand itself because this would cause the overlayed musical notes to move with the playing hand. The virtual musical score disappears whenever the pianist looks up from the keys.

Footprint: Another step in personal context

Footprint, our second personal-context application, uses the feet as the focal point for computer assistance. Previous work on foot-based user interfaces can generally be classified as hardware-based or vision-based implementations. Applications for such interfaces include dance performance, choreography, motion capture for animation, and interactive entertainment.

Hardware-based schemes often rely on body-mounted magnetic, ultrasonic, or LED devices that monitor the motion of the whole body. Hardware systems can quickly provide great accuracy and a wealth of data but require complex infrastructure or equipment. Computer vision systems make use of a single camera or several cameras fixed in the environment to monitor a specific location for body motion. While some systems rely on body-placed markers to aid visual detection, many analyze the scene with only an a priori model of the human body.[11] These systems are more interested in entire body motion rather than just foot motion. One exception to this rule is a technology[12] that derives 3D motion data from a bicyclist's legs by analyzing specially textured shorts. Computer vision systems often free the users from having to wear any special devices, but they also require good lighting conditions and fast computers to process complex algorithms.

Footprint operates on the same minimal wearable computer system as Handel: a small laptop with a see-through head-mounted display complete with an attached video camera. The user's feet trigger computer interaction when Footprint detects them in-screen. Footprint accomplishes foot detection by analyzing the frames captured by the video camera and exploiting a priori knowledge of the owner's feet.

Footprint can handle basic waltz steps. Figure 3 shows Footprint's architecture. A typical practice session begins when the user starts the application and loads the system settings and dance information. The user activates an internal timer, which allows Footprint to synchronize dance steps to time. The user then performs the dance to music the computer supplies. Whenever the user needs help, he or she simply looks down. As Figure 4 shows, Footprint then presents graphics and text that indicate where the feet should move next. This information disappears when the user looks back up. Looking down at the feet provides a natural means to interact with the computer. As in Handel, Footprint only shows information when the user needs it, which minimizes graphical clutter on the limited-resolution head-mounted display.

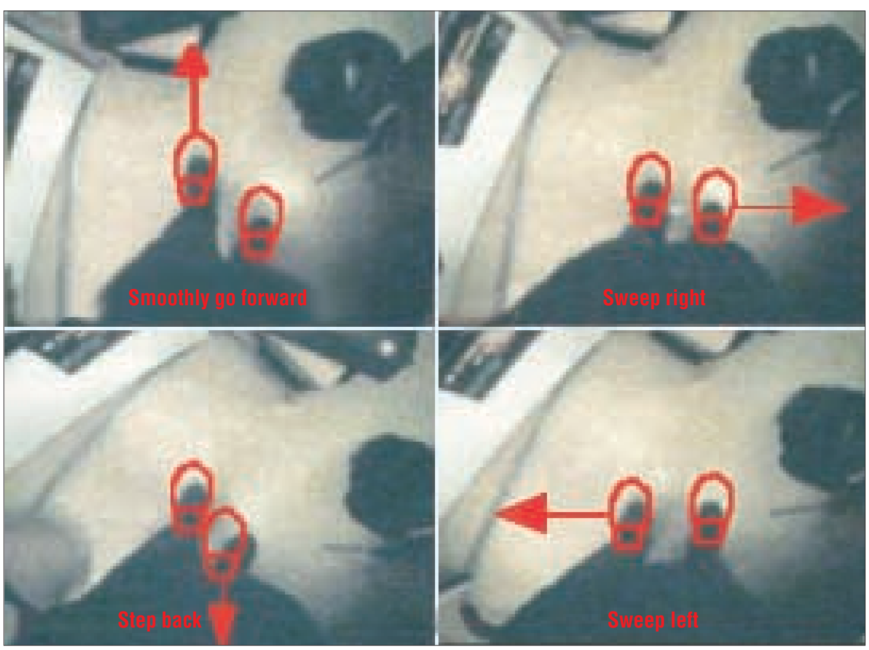


Figure 4. Dance step instructions as seen by the user's head-mounted display.

The feet-detection algorithm assumes that the user is wearing dark shoes and that the floor is fairly uniform in color. Presuming that lack of edges corresponds to uniformity, the algorithm first checks for a fairly uniform background by using a nonlinear spatial activity detector. If the detector senses a cluttered background, Footprint assumes the user is not looking at a uniform floor and will not perform any foot detection.

If the current video frame passes the floor test, Footprint matches a predefined shoe template against a coarse grid on the current frame. The grid is set to the left half of the image to search for the left foot. At each grid position, Footprint equalizes the local rectangular region to be compared against the template and calculates the local variances inside and outside the shoe area. If the difference falls below a threshold (indicating the texture inside and outside the shoe is the same), or if the total difference within shoe area against the template exceeds a threshold (indicating the shoe area does not have a dark shoe), then Footprint does not detect a foot.

Otherwise, Footprint computes a measure proportional to the match against the template divided by the difference of variances. Footprint classifies the grid position with the smallest measure (that still falls under a threshold) as a foot. Footprint then repeats the process to find the right foot, except that Footprint sets the grid to the right of the discovered left-foot position. Figure 5 illustrates the foot-detection algorithm under different lighting and floor conditions. On subsequent steps after the first, the system searches around the last detected coordinates first before performing a full grid search.

We've represented the dance itself as an XML text file using custom markups. As Figure 6 illustrates, Footprint represents the dance moves clearly. Footprint can present these moves in sequence using common ballroom dance step speed denotations such as "quick" or "slow" along with text descriptions of each movement. This new "dance markup language" is similar to SMIL, a markup language for synchronized multimedia.[13] All the parameters controlling Footprint are stored in another XML text file. Footprint runs at about four frames per second on a Pentium 233 laptop, which includes all the image processing, video capture, and graphics rendering required by the ballroom dancing task. It detects the feet well and runs effectively with the basic waltz.

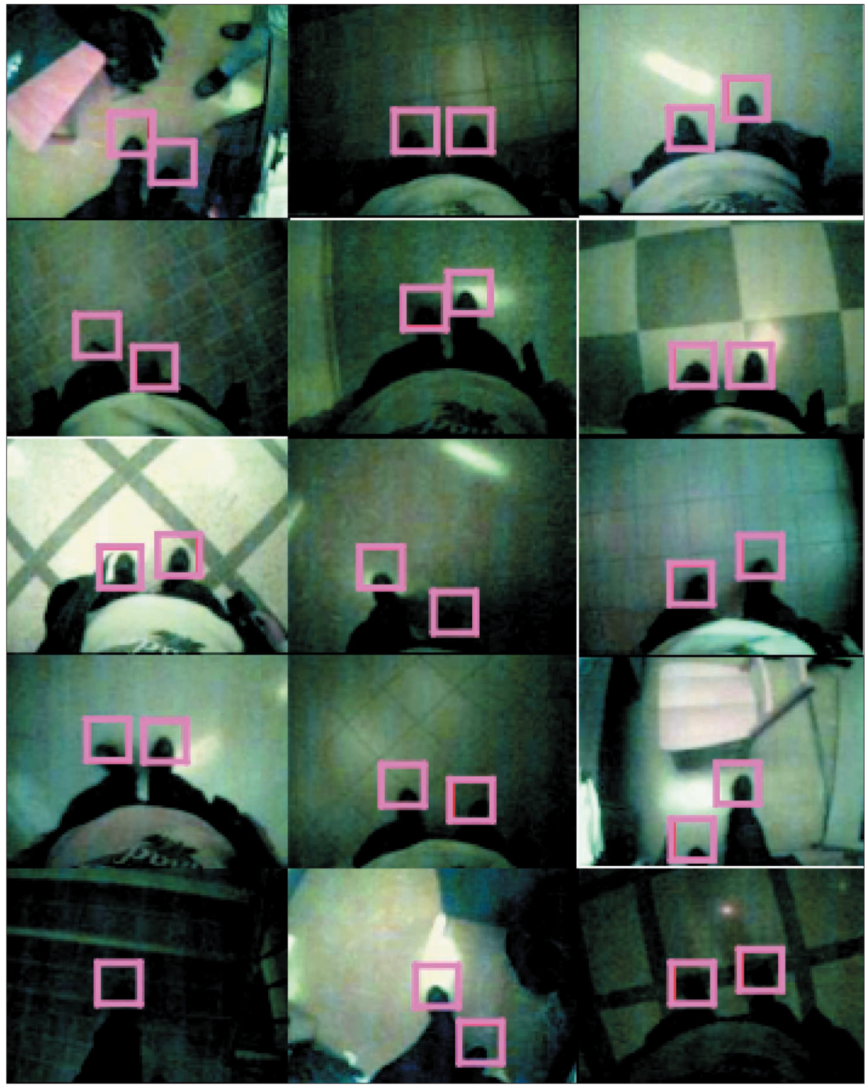


Figure 5. Footprint's foot detection in different lighting and floor conditions, as seen by the head-mounted camera. Detected feet are highlighted by rectangles.

Future directions

We tested Handel successfully only on an acoustic piano for a short musical piece. While the system proved to be comfortable to use, there are of course numerous improvements that could be made, and we would like to do a formal study to assess Handel's benefit (or detriment) to memorizing piano music. Handel's FFT phase approach combined with skin detection seems to be sufficient for detecting a hand and for determining which hand is currently in focus. An improvement would be to employ projective-based scene analysis technology such as the kind found in wearable camera systems.[1]

Footprint would likely benefit from a faster computer, foot-pose recognition, and additional user tests to optimize the dance instruction presentation. It would be interesting to study other modes of computer-assisted teaching for Footprint, such as having Footprint measure feet movement to assess proper steps. Extending the system to recognize and coordinate with a live partner would also be desirable. Using an XML-based dance step file to represent content and an XML configuration file as a style sheet means that Footprint is, in one sense, a browser for wearable computer interfaces.

Because the dance markup language is a simple description of dance steps, it can be interpreted for different purposes on other platforms. For instance, another wearable computer could create XML-based data from

```
<dance>
   <title>Basic Waltz</title>

   <step name="advance" duration="quick">
      Smoothly go forward
      <leftfoot direction="forward">Left first
         </leftfoot>
   </step>

   <step name="right" duration="quick">
      Sweep right
      <rightfoot direction="right">Right first
         </rightfoot>
   </step>

   <step name="right wait" duration="quick">
      Close
      <leftfoot direction="hold">Left arrives late
         </leftfoot>
   </step>

   <step name="back" duration="quick">
      Step back
      <rightfoot direction="back">Right first
         </rightfoot>
   </step>

   <step name="left" duration="quick">
      Sweep left
      <leftfoot direction="left">Left first
         </leftfoot>
   </step>

   <step name="left wait" duration="quick">
      Close
      <rightfoot direction="hold">Right arrives
         late</rightfoot>
   </step>
</dance>
```

Figure 6. The dance markup file for the basic square-step waltz.

streaming sensor data. A 3D-capable XML desktop browser could translate the dance step file into a dancing avatar that could be incorporated into a virtual reality environment or a computer graphics movie. Online XML database engines could index and catalog the dance step file in a repository, allowing for text-based searches for human gesture and motion.

In general, context-aware applications can exploit XML as a foundation to create readable, portable, and indexable notations for human gesture, motion, and interaction with the real world. Because gesture, motion, and interaction vary over time and depend on different conditions, context-aware notations might adapt properties and behaviors from scripting languages and temporal-based notations.

With only simple computer-vision techniques, Handel and Footprint demonstrate the great possibilities for more natural human–computer interaction. And ,using XML in Footprint illustrates the potential for XML to become a portable format to represent human activity for both wearable and desktop applications. ■

References

1. S. Mann, "Wearable Intelligent Signal Processing," *Proc. IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2123–2151.
2. T. Starner, B. Schiele, and A. Pentland, "Visual Contextual Awareness in Wearable Computing," *Proc. Second Int'l Symp. Wearable Computers*, IEEE CS Press, Los Alamitos, Calif., Oct. 1998, pp. 50–57.
3. R.T. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, no. 4, Aug. 1997, pp. 355–385.
4. S. Feiner, B. MacIntyre, and D. Seligmann, "Knowledge-Based Augmented Reality," *Comm. ACM*, July 1993, pp. 53–62.
5. M. Billinghurst, H. Kato, and I. Poupyrev, "The Magicbook: Moving Seamlessly Between Reality and Virtuality," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, May/June 2001, pp. 6–8.
6. M. Billinghurst et al., "An Evaluation of Wearable Information Spaces," *Proc. IEEE Virtual Reality Annual Int'l Symp.*(VRAIS 98), IEEE Press, Piscataway, N.J., Mar. 1998, pp. 20–27.
7. A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* IEEE CS Press, Los Alamitos, Calif., Jan. 2000, pp. 107–118.
8. M.M. Krueger, "Environmental Technology: Making the Real World Virtual," *Comm. ACM*, July 1993, pp. 36–37.
9. J. Miyasato, "See-Through Hand," *Proc. 8th Australian Conf. on Computer Human Interaction (OzCHI 98)*, IEEE CS Press, Los Alamitos, Calif., November 1998.
10. L. Cheng and J. Robinson, "Dealing with Speed and Robustness Issues for Video-based Registration on a Wearable Computing Platform," *Proc. Second Int'l Symp. Wearable Computers*, IEEE CS Press, Los Alamitos, Calif., Oct. 1998, pp. 84–91.
11. J. Ohya, J. Kurumisawa, and R. Nakatsu, "Virtual Metamorphosis," *IEEE Multimedia*, vol. 6, no. 2, April/June 1999, pp. 29–39.
12. F. Lerasle, G. Rives, and M. Dhome, "Tracking of Human Limbs by Multiocular Vision," *Computer Vision and Image Understanding*, vol. 75, no. 3, Sept. 1999, pp. 229–246.
13. W3C World Wide Web Consortium, "Synchronized Multimedia," www.w3.org/AudioVideo (current 18 June 2001).

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Authors

Li-Te Cheng is a research scientist at Lotus/IBM Research. His research interests include collaboration for wearable and ubiquitous computing, virtual reality, and computer-vision–supported augmented reality. He is completing a PhD in Electrical Engineering at Memorial University of Newfoundland. He received an MASc and a BASc in Systems Design Engineering from the University of Waterloo and a Certificate of Space Studies from the International Space University. He is a member of the IEEE Computer and Signal Processing societies and ACM SIGCHI. Contact him at li-te_cheng@lotus.com.

John Robinson is Professor of Media Technology in the Department of Electronics at the University of York. From 1996 to 2000 he held an Industrial Research Chair at Memorial University of Newfoundland, where he performed the work reported in this article. Previously, he worked for Standard Telephones and Cables Ltd., Basildon, UK, Bell-Northern Research Ltd., Verdun, Quebec, Canada, and the University of Waterloo, Ontario, Canada. He received postgraduate degrees in Electronic Engineering, from the University of Essex, and in the Humanities from Memorial University of Newfoundland. He is a Fellow of the IEE and a Member of IEEE. Contact him at www-users.york.ac.uk/~jar11/.

Telme: A Personalized, Context-Aware Communication Support System

Kaoru Sumi, *Communications Research Laboratory*
Toyoaki Nishida, *University of Tokyo*

True global communication will require more than just language translation technologies. To fully understand each other, people also need context-specific information. The authors have developed Telme, a support system that gives users real-time information to help bridge the knowledge and experience gap.

As communications become more global and instantaneous, we have an increasing opportunity to converse with people who live in different cultures, work in different fields, and speak different languages. New communication devices are now in development, including wearable computers and real-time speech translation systems.[1]

Even with such translation systems, however, people will still have difficulty understanding each other if they do not have similar background knowledge or experiences.

To help remedy this, we developed Telme, a communications support system that acts as a mediator between people with varying levels of knowledge and experience. Telme supports real-time communications by presenting information from a knowledge base customized according to the user's profile and operational records.

We model the user's knowledge structure based on two premises. The first is that users' knowledge can be deduced from the questions they ask. In the field of cognitive science, listening to questions is one way of determining how much one knows about a certain subject.[2,3] In our research, this lets us determine what users know without placing a burden on them.

The second premise is that different people can understand the same topic from various viewpoints and use different words and concepts to describe the same thing. We thus use the idea of a "conceptual space"—simple, extensible sets of related keywords—to represent concepts in our system. Although the conceptual space largely depends on the person, some concepts are partially reusable. Cognitive scientists and philosophers have used similar methods to model the human concept-formation process.[4,5]

Telme works by inferring the conversation context based on defined conceptual spaces and the user's own conceptual space, which it generates from the user's profile and his or her system interactions. The system then uses this information to give users "assistant information" about the current topic and its context.

We implemented Telme in two domains—cooking and gardening—and tested the system with 21 subjects, all of whom were computer literate but had varying degrees of domain knowledge. Based on these tests, we analyzed both the effectiveness and appropriateness of Telme's information for providing real-time communications support.

Framework

Figure 1 shows the Telme framework for wearable computers connected to a central knowledge base server. The server controls a background knowledge database and downloads data on user request. The computer's display shows four windows:

- the main window, which presents the speaker's dictated words on screen;
- the knowledge conceptual space, which shows the listener's knowledge space;
- the context conceptual space, which shows information about the topic; and
- the assistant window, which shows text and pictures to explain the speaker's words.

Figure 2 shows the wearable computer unit. Telme also works for TV-like broadcasting on a nonportable

1094-7167/01/$10.00 © 2001 IEEE

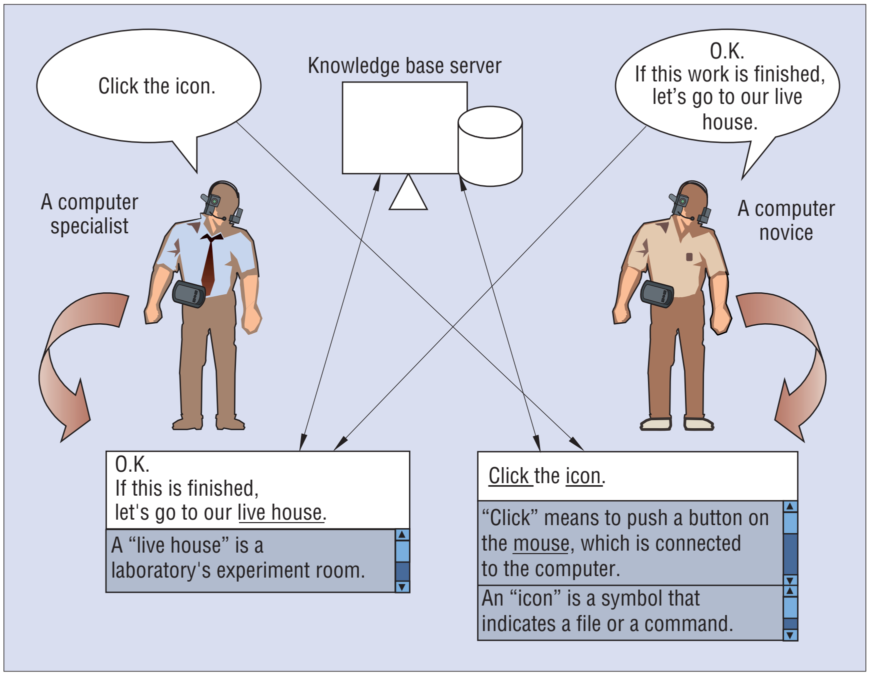


Figure 1. The Telme framework. The wearable computer's display presents speech as dictation in the main window. The assistant window shows text and pictures that describe the speaker's words. Two other windows show additional information: the context conceptual space shows information on the topic and the knowledge conceptual space shows user knowledge of the topic.

Figure 2. Telme works on wearable computers such as the one shown, as well as on desk and laptop computers.

Before using the system, all users must select their domain specialty from among several conceptual spaces. These domains then serve as user profiles.

unit, such as a desk or laptop display.

Presenting assistant information

Assuming that user s is the speaker and user l the listener, we present user l's annotated information in the assistant window $A(l)$ as

$$A(l) = T(d, s) \cdot F(l),$$

where $T(d, s)$ represents the information transformation according to both the topic domain and user s's background knowledge and $F(l)$ represents what user l doesn't know. That is, Telme calculates the possibilities of each word's domain based on the current topic domain (d) and user s's background knowledge. From this, the system infers the conversation context and, if the word has several meanings across domains, selects the best meaning. Thus, Telme conveys information about the ongoing topic based on user s's intent and filters it into different information based on what user l doesn't know. When Telme recognizes the speaker's words, it automatically displays information that the listener doesn't know about the current topic domain and the speaker's domain in the assistant window.

Presenting information

Telme's prepared information consists of a knowledge base and defined conceptual spaces. The *knowledge base* consists of terms and question–answer pairs corresponding to each term. The answer part of each question–answer pair is presented as text or pictures, and three kinds of questions are used:

- "*What*," which requests an explanation of the word's used;
- "*Example*," which requests an example of the word's meaning; and
- "*Why*," which requests the reason for using the word (mainly verbs).

A *conceptual space* consists of keywords defined by their interrelationships. Domain specialists design individual conceptual spaces based on their own viewpoints. For Telme, we designed simple conceptual spaces using documents created by domain specialists. For example, in the gardening domain, we might link two vegetables, such as "tomato" and "eggplant." Such relationships are reversible, and therefore Telme can determine the user's knowledge based on either keyword.

Inferring user knowledge and conversation context

Telme infers the user's knowledge based on the user's own knowledge conceptual space. The system generates this space based on the conceptual spaces of the specialists and the information space of the current topic; it adapts the space to the user based on his or her operations (such as questions, information checking, and so on).

Nodes in the knowledge conceptual space represent concepts, and links represent relationships between concepts. Both the nodes and links have weight values. The system weights the nodes based on the user's knowledge level and weights links based on the relationships between the linked concepts. Telme assumes that when users question a concept, it is something that they don't know, and weights it and related concepts accordingly. Similarly, when users erase a concept by closing a window, the system assumes user knowledge of it and related concepts.

The system infers users' knowledge level by calculating the weights of concept-related links. Users can also select useful information by clicking a checkbox, which lets them review and print out the information. The system stores these operational records and uses them to adapt to the user by adjusting the weights of links in the knowledge conceptual space.

How Telme works

First, to extract data from conceptual spaces and apply it to the user's own conceptual space, the system assigns weight values reflecting the listener's background knowledge to all nodes in the knowledge conceptual space. Based on the user profile, the system gives a node maximum value when the listener has knowledge of it and minimum value when the user does not.

For example, when user α queries *Keyword b* of *What*, the system obtains keywords linked to *Keyword b* (such as keywords *n*, *o*, *p*, *x*, and *y*) from the relationship between the number of conceptual spaces in user α's conceptual space. If no links exist in user α's conceptual space, the system gives the keywords initial values based on the values of either the current topic domain *d* or the speaker's background knowledge *s*, whichever is greater. For the links existing in user α's conceptual space, it uses the weights they have at that time.

Next, the system selects information to present. If it finds information that can be presented in the current context, it determines a maximum of five keywords based on the keyword probabilities. When the user drags *Keyword x* and selects *What,* it shows the answer to *Keyword x* of *What*.

Finally, the system updates the user's conceptual space. The user can indicate in the assistant window whether the current information is important or not either by clicking the checkbox or closing the window. For example, if the user clicks the checkbox to save the textual explanation or picture in the assistant window, the system assumes that the explanation is important to the user. If the user closes the assistant window, the system assumes that the explanation is not important.

If user α saves the explanation for *Keyword x* of *What*, the system rewards *Keyword b* ↔ *Keyword x*, which are tracks of inferred keywords. At the same time, the system searches for all conceptual spaces. If a link in the user's conceptual space has the same keyword on either side of *Keyword b* ↔ *Keyword x*, the system rewards the link in the conceptual space. It then updates the user's conceptual space by using conceptual spaces of, for example, specialists B and C because they contain links to *Keyword b* ↔ *Keyword x*. The link to *Keyword b* ↔ *Keyword x* itself is rewarded for updating the conceptual space. Because the links are reversible, it also rewards *Keyword x* ↔ *Keyword b*. Likewise, if a link with the same keyword as that to the left of a rewarded link is available—for example, *Keyword b* ↔ *Keyword y*, *Keyword x* ↔ *Keyword q*, or *Keyword x* ↔ *Keyword s*—it is also rewarded (see Figure 3). If the user indicates that *Keyword x* of *What* is unimportant (by closing the screen), the system lowers the link weights accordingly.

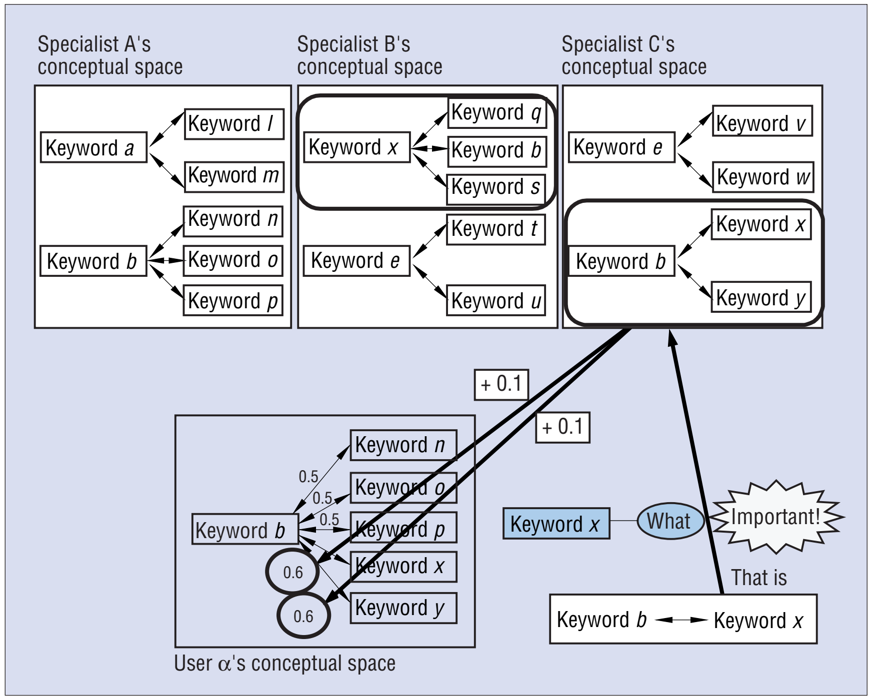


Figure 3. To generate the knowledge conceptual space, the system rewards links related to user operations by increasing or decreasing their weights.

The system assumes that *Keyword b* is in an unknown concept because the user questioned it. The system gives the node a minimum value, then calculates the value of related concepts by multiplying the nodes' weight values by the values of the links to them. If the user erases the concept, the system assumes that the user knows it and gives the nodes maximum values, and (as with the previous example) calculates the values of related concepts by multiplying the nodes' weight values by the values of the links to them. This repeated weight adjustment of keyword relationships corresponds to collecting several people's viewpoints.

To infer the current topic domain, we use a much simpler method based solely on determining word frequency in a specialist's conceptual space.

Presenting the conceptual spaces

The knowledge conceptual space shows the user's own conceptual space and how much knowledge the user has. The context conceptual space shows the current topic and a corresponding specialist's conceptual space.

Telme visually presents conceptual spaces to users in a 2D space by matching the relationships in pairs of keywords to spatial distances using a multidimensional scaling method.[6] By simply glancing at these spaces, users can easily understand the knowledge structure.

By looking at the knowledge conceptual space, users can see which parts of the knowledge they know or don't know when the assistant window automatically appears. Concepts in the context conceptual space show users which concepts are related to the current topic. By comparing these two conceptual spaces, users can understand the difference between their concepts and those of a specialist.

Implementation

We implemented the system in two different domains—cooking and gardening—and evaluated it in conversations between people with different knowledge levels. For voice dictation, we used the commercial software application Via Voice by IBM. We installed the system on a Pentium III 600-MHz PC and used the Visual Basic 6.0 programming language.

Figure 4 shows a Telme screen (adapted from the original Japanese version) comprising the four windows. In this case, the system is facilitating a discussion about cooking between a novice and a specialist. The specialist (the speaker) is advising the novice (the listener) about ingredients for minestrone. The dictation area (at the top of the screen) shows the speaker's statement, "I always use garlic, onions, leeks, celery, carrots, kidney beans, potatoes, tomatoes, olive oil, bouillon, and Parmesan."

There are five assistant windows below the dictation area; each defines a word that the system infers the user doesn't know (olive oil, celery, and Parmesan). The lower right shows the context conceptual space, which contains keywords related to the current topic (cooking). The related words ("carrot," "potato," and so on) are shown as dark colored icons so that users can easily distinguish them from other icons. Other terms are arranged according to the relationships between each combination of terms in the field. For example, the vegetables "carrot," "potato," and "onion" are in a group, as are "bouillon" and "Parmesan," which we categorize as Italian seasonings. Herbs are in another group.

In the knowledge conceptual space (upper-right corner), the term icons are colored according to weights based on the user's knowledge. The known-knowledge space (selected) shows words that the user knows ranked by priority, with the darkest indicating well-known words and the lighter terms indicating potentially unknown words. Alternately, the user can select the unknown-knowledge conceptual space and the system will prioritize unknown terms.

Based on the user's operational history and user profile, the system makes several inferences. First, using the user's operational history (a conversation about onion gratin soup), the system infers that the user knows "onion" and "garlic," and as a result, does not explain these or related terms ("onion," "garlic," "carrot," "kidney bean," "potato," and "tomato"). The system also infers that the user may not know the meaning of "leeks," "celery," "olive oil," and "Parmesan," based on the user's questioning of the word "leeks."

Figure 5 shows a screen from the gardening domain (also adapted from the Japanese version) where the specialist (the speaker)

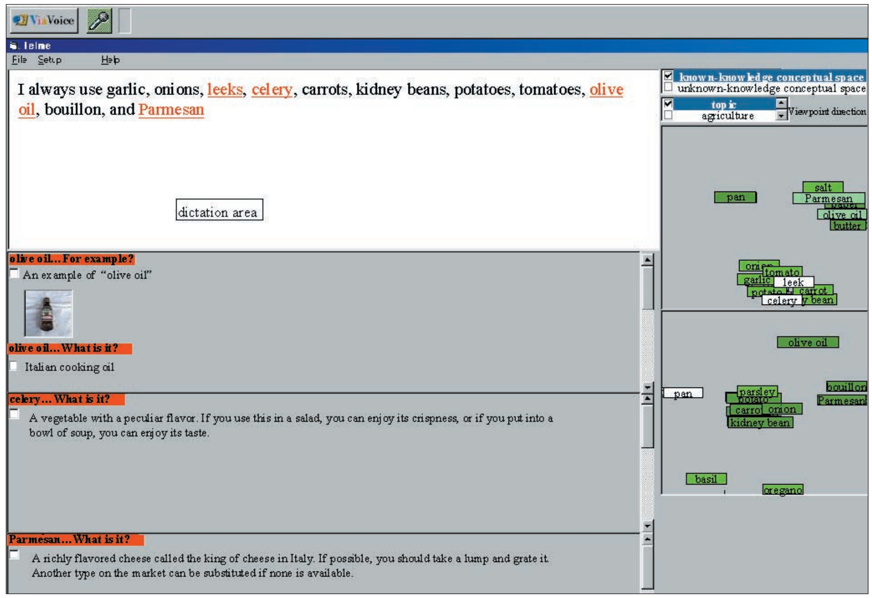


Figure 4. A screen shot of the Telme system. In this example, a novice and an expert are discussing cooking. Dictation appears in the large upper box, with five assistant windows below it (three are shown here). On the right are the knowledge conceptual (upper right) and context conceptual (lower right) spaces.

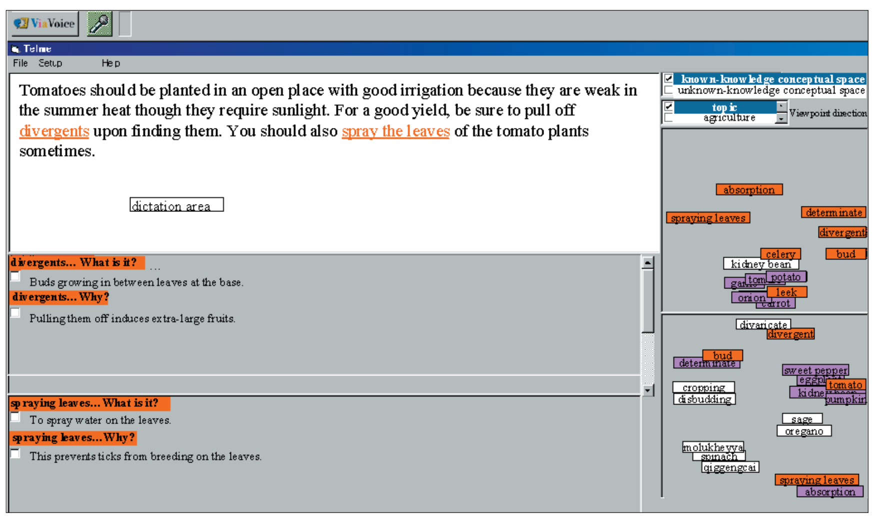


Figure 5. A Telme screen shot from a gardening-related conversation.

advises the novice (the listener) on how to care for tomatoes. The dictation area features several underlined words. Based on the user's questioning of the term "determinate" (having growth where a bud or flower terminates at the growing tip), the system infers that the user doesn't understand the related terms "divergent" and "spray leaves." These words are explained in the assistant windows below. When the user clicks on the checkbox in the assistant information window, the specialist's advice can be recalled or printed out later. The system tracks information that the user selects and updates the user's knowledge conceptual space accordingly.

The context conceptual space features several groups, including a group of herbs ("oregano," "sage," and "basil") and a group of synonyms related to growth promotion via pruning ("cropping" and "disbudding"). Finally, based on the conceptual space's contents, the user can assume that "divergent" is a concept related to "bud" and "determinate."

As Figures 4 and 5 show, several terms in the cooking and gardening domains (especially the names of vegetables) overlap.

However, the system treats the terms differently. It presents different explanations for each domain by inferring the current topic and the speaker's specialty. Words related to the terms are also different in each domain, even if the terms are the same (for example, a group of ingredients for Italian cooking would be different than a grouping of leaves and fruits in the gardening domain).

Experiment and evaluation

To evaluate Telme, we performed an experiment using 21 university graduate students and secretaries who had expertise in various subjects (ranging from system science to education and psychology). We selected people who were familiar with computers to circumvent computer literacy issues.

Because we were mainly interested in evaluating how the system adapts, we prepared small knowledge bases and conceptual spaces. We prepared the scenarios and conversation topics in advance. We asked the subjects to behave as if they were participating in an ordinary conversation and to let the conversation progress naturally. We did not use speech recognition devices. The subjects used the system to read the conversational sentences and assistant information it presented.

The first topic was cooking, followed by gardening. We asked the subjects to continue using Telme until all sessions in the conversation had finished. In our analysis, we focused on the effectiveness and usefulness of the assistant information as well as its appropriateness for real-time communications support.

We analyzed the users' operational histories and the questionnaires we asked them to fill out before and after the test. The questionnaires asked users to rank their impression of the system on a scale from 1 to 5 (5 = excellent, 4 = good, 3 = passable, 2 = not so good, and 1 = bad). We considered the values of 3 or greater to be positive. To gauge the reliability of the answers, we asked the subjects to give reasons for their responses on most of the questions.

Effectiveness of assistant information

When asked to evaluate the effectiveness of the assistant information, 86 percent of the subjects said Telme presented new information, and 90 percent said the presented information was useful for understanding the topic. Out of an average of nine instances when assistant information was offered, 2.89 instances were deemed useful. As to relevance, 80 percent of the subjects said Telme's assistant information was related to the current topic. Other findings include the following:

- 80 percent said that the knowledge conceptual space presented the relationship between concepts well.
- The majority said that Telme's assistant information was compatible with their knowledge level.
- 85 percent said that the context conceptual space presented relationships between concepts well.

In the latter case, one subject said that the icons for the current topic (cooking ingredients) were well clustered in the conceptual space. Another subject said that it was easy to find the conceptual meanings of unknown words by finding known words that were close to them in the conceptual space.

We divided the subjects into groups according to domain knowledge levels and analyzed the effectiveness of presenting assistant information in each group. We found that as the knowledge level increased, more subjects were inclined to think that the information presented was relevant to both the topic and their knowledge level. This might be because their advanced understanding of the domain led to greater overall understanding of the content.

Value for real-time communications support

We analyzed the appropriateness of the information in terms of real-time communications support, using both the questionnaires and records of the subjects' system interactions. We first analyzed users' real-time system interaction (such as selecting important information, erasing useless information, and questioning).

All subjects said that they were not reluctant to click the checkbox to save the information, and 85 percent of the subjects said they were willing to use this operation, primarily to reuse and verify useful information. Also, 85 percent said they were willing to click the checkbox to close the window, which erases information. In the comments, most subjects said that erasing was necessary because they became confused when there was too much information on screen. Some subjects gave negative comments, saying that they felt reluctant to discard information that they might need later.

Nearly all subjects (95 percent) said that they would drag unknown terms to the dictation area, then select a question to get more information about the terms. The main reason they gave for their willingness to do this was that it was a quick and easy way to get an answer. One subject said that this operation was especially convenient when she knew something about the question, but was not completely sure of its meaning.

Once the conversation began, on average the user executed the first operation in 0.25 seconds. From this, we conclude that the operation would not necessarily interrupt a conversation. However, 62 percent of the subjects said that they felt the system was slow. Much of this is related to the construction of the conceptual spaces; the assistant information is presented automatically and processed instantaneously. Because Telme uses a multidimensional scaling method to visualize the conceptual spaces, the processing cost is high. However, this is only a problem with the visualization process itself. We can completely solve the problem by processing the visualization individually. For this reason, composing and displaying conceptual spaces should be a background process.

Usefulness for different types of communications support

To explore future possibilities for Telme, we asked for users' impressions about several possible communication devices, including wearable computers. Using the subjects' questionnaires, we also analyzed the general usefulness of presenting assistant information for communications support.

Our analysis captured a peripheral view of the system's support for

- *Computer chat.* All subjects agreed that Telme would be a useful computer-chat support system (19 percent checked "5" and 57 percent checked "4"). Among the positive comments were, "We may find it convenient if we confront something unfamiliar," and "We can talk with someone without the conversation being interrupted." There was also a negative comment: "We may prefer to ask a question directly if it is brief."
- *Generic interviewing.* 85 percent of the subjects agreed that Telme would be useful as an interview support system (19 percent checked "5" and 37 percent checked "4"). Among the positive comments were those saying that Telme is very convenient, that it can be used while cooking, and that

Related Work

In the future, communication is likely to become more spatiotemporal, global, and instantaneous. Presenting real-time and personalized information by computer can help people understand each other instantaneously as well. Such technology can also enrich information content.

Current research on using tagging technology to enrich WWW content has received a lot of attention. Semantic Web Development (www.w3.org/2000/01/sw/DevelopmentProposal) is one such project. We believe that helping users better understand a particular topic requires both richer content and the addition of contextual information. Another effort to enrich WWW content is that of Nagao, Shirai, and Squire,[1] but it is not an automatic adaptation system nor does it account for differences in knowledge levels among users.

Bradley Rhodes developed a system that provides information that might be relevant to the context.[2] The system uses a wearable computer and a database of past descriptions (such as memos) as information sources. However, based on our research related to human understanding, our desire is to have much richer and more personalized information for different people.

Because Telme adapts information to the user and the context, representing meaning structures is important. Researchers have proposed several question–answer systems for "know-how" knowledge that aim at information distribution among several presenters and users. For example, Answer Garden is based on a relationship network tuned by the user's questions to support the user's knowledge acquisition.[3] If detailed meaning structures can be designed, good results can be generated.

However, because question–answer systems require the information spaces of multiple presenters, such systems require multiple, detailed meaning structures. Building these structures is expensive, and, once built, they cannot be modified easily. Because a meaning structure tends to be biased by the viewpoint of the person who gives it its meaning, the system can only consider specific user concepts; it cannot consider each user's concepts individually. The use of meaning structures can therefore improve the system performance only in a general way, and not for individual users.

We propose a system that uses knowledge-space structures as simple related links. With these structures, the system can generate the user's conceptual space while taking into account his or her individual viewpoints and existing concepts from built-in conceptual spaces.

We previously designed an information navigation system called Takealook.[4] The system infers the user's interests by generating a conceptual space of his or her interests from other users' defined conceptual spaces. With Telme, we used the same method because it can be used to infer knowledge. Although Telme considers only just-in-time adaptation to the current context, the method is flexible because conceptual spaces simply consist of keywords and weights, and the weights can be easily adjusted according to the inferred context.

References

1. K. Nagao, Y. Shirai, and K. Squire, "Semantic Annotation and Transcoding: Making Web Content More Accessible," *IEEE Multimedia*, vol. 8, no. 2, Apr.–June 2001, pp. 69–81.
2. B. Rhodes, "The Wearable Remembrance Agent: A System for Augmented Memory," *Personal Technologies*, vol. 1, no. 4, Mar. 1997, pp. 218–224.
3. M.S. Ackerman and T.W. Malone, "Answer Garden: A Tool for Growing Organizational Memory," *Proc. ACM Conf. Office Information Systems*, ACM Press, New York, 1990, pp. 31–39.
4. K. Sumi, "Intelligent Tool for Facilitating Creative Communication," *Dynamic Knowledge Interaction*, Toyoaki Nishida, ed., CRC Press, Boca Raton, Fla., 2000.

it provides accurate explanations. A negative comment was that the system might not be as useful as a chat system, wherein users can question each other directly during the conversation.

- *Interviewing people from different fields.* All the subjects agreed that Telme would be useful in this area (29 percent checked "5" and 47 percent checked "4"). Among the positive comments was that the system makes it convenient for the user to learn a word's meaning in the speaker's field, particularly when the meaning might vary across fields. Another positive comment was that a user can learn a word's meaning by questioning the system without interrupting the conversation, noting that this would be especially helpful when the user was the only person in a group who did not know a word. A negative comment was that it is impolite to use a computer when interacting with other people.
- *Real-time conferencing.* 95 percent of the subjects agreed that Telme would be useful as a real-time conference support for meetings between multiple people (24 percent checked "5" and 52 percent checked "4"). Among the positive comments was that the system gives the user an opportunity to quietly analyze unfamiliar words without interrupting the conversation. Another subject said that it might also be useful for checking unfamiliar words when attending a lecture. A negative comment was that people might stop listening to one another and become overly reliant on the machine.
- *Broadcasting.* All the subjects agreed that Telme is useful as a broadcast support system (76 percent checked "5" and 19 percent checked "4"). Among the positive comments were that it would be convenient to immediately understand unfamiliar political terms, and that the system might be useful for constructing knowledge and checking concepts related to the current topic.

Given these results, we conclude that Telme leaves a good impression for any communication style. In the opinions of users—who were merely considering different communication modes—Telme seems more useful as a computer chat system than as an interview support system, more useful as a conference system than an interview support system, and generally useful as a broadcast support system. In analyzing user comments, we found that they considered the system most effective when they could not question others directly—for example, when the user might be the only one who didn't understand a word or didn't want to interrupt others to ask a question.

Using Telme, even novice users can easily grasp unfamiliar things or avoid misunderstandings by glancing at the explanations. Telme's assistant information provides users with useful knowledge and a deeper understanding of the topic. As a real-time communication support system, Telme is relatively fast and easy and operates without interrupting the flow of conversation.

Because Telme is a customizable medium, it can help overcome the differences in knowledge among people. Personalization according to context will become increasingly important as human networks expand and various kinds of new media appear in the near future. Although we can't foresee what kinds of new communications media will emerge, we believe Telme will be effective for real-time and global communications between people from different fields, cultures, and languages.

In the current version of Telme, we collected most of the information manually, which took much effort. In addition, creating and updating the knowledge base and conceptual spaces were costly endeavors. We need a mechanism to enable inexpensive information collection. Developing such a mechanism will require further improvements in document-tagging technologies that deal with semantic structures. ■

References

1. T. Takezawa et al., "A Japanese-to-English Speech Translation System: ATR-MATRIX," *Proc. Int'l Conf. Spoken Language Processing* (ICSLP '98), Causal Productions, Adelaide, Australia, 1998, pp. 957–960.
2. A. Flammer, "Towards a Theory of Question Asking," *Psychological Research*, vol. 43, 1981, pp. 407–420.
3. N. Miyake and D.A. Norman, "To Ask a Question, One Must Know Enough to Know What Is Not Known," *J. Verbal Learning and Verbal Behavior*, vol. 18, 1979, pp. 357–364.
4. R.A. Finke, T.B. Ward, and S.M. Smith, *Creative Cognition*, MIT Press, Cambridge, Mass., 1992.
5. M. Polanyi, *The Tacit Dimension*, Routledge and Kegan Paul Ltd., London, 1966.
6. Y. Sumi, K. Hori, and S. Ohsuga, "Computer-Aided Thinking by Mapping Text-Objects into Metric Spaces," *Artificial Intelligence*, vol. 91, no. 1, Mar. 1997, pp. 71–84.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Authors

Kaoru Sumi is a research associate at Osaka University's Institute of Scientific and Industrial Research. She worked on Telme as a researcher for the Breakthrough 21 Nishida Project at the Communications Research Laboratory. Her research interests are knowledge engineering, communication support systems, user modeling, and human–computer interaction. She received her BS in physics from the Science University of Tokyo, her MS in systems management from the University of Tsukuba, and her PhD in engineering from the University of Tokyo. Contact her at the Institute of Scientific and Industrial Research, Osaka Univ., 8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan; kaoru.sumi@acm.org; www.ei. sanken.osaka-u.ac.jp/members/kaoru/index.html.

Toyoaki Nishida is a professor in the Department of Information and Communication Engineering, School of Information Science and Technology, the University of Tokyo, and leader of the Breakthrough 21 Nishida Project at Japan's Communications Research Laboratory. The project aims to understand and assist networked communities. His general research interests are in artificial intelligence; his current research focuses on community computing and support systems, including knowledge sharing, knowledge media, and agent technology. He is an area editor in intelligent systems for *New Generation Computing* and on the editorial boards of *Autonomous Agents and Multiagent Systems*, *Knowledge and Information Systems*, and the *International Journal of Knowledge-Based Intelligent Engineering Systems*. He received his BE, ME, and doctorate of engineering from Kyoto University. Contact him at the Dept. of Information and Communication Eng., Univ. of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.

A Headset-Based Minimized Wearable Computer

Soichiro Matsushita, *Toyo University (R&D Center, Toshiba)*

A headset with a sensory system and a short-range wireless radio transceiver can become a highly available, context-aware peripheral device. The author describes a headset that uses a combination of low-power motion sensors and microcontrollers to translate a user's behavior into corresponding symbolic codes.

Who wants to wear a small personal computer all the time? If you don't want to experience tremendous heat due to power dissipation in some PCs, you have to endure low performance, and although notebook PCs could be worn all the time, heavy and incredibly short-life batteries make them impractical. Even the tremendous progress in silicon integrated circuits doesn't help because the performance gap between a state-of-the-art desktop machine and a wearable one doesn't disappear. So, we have to find wearable features that a conventional PC and its derivatives do not have.

Requirements for hardware in terms of size, weight, power consumption, and allowable heat radiation have severely restricted the computational performance of wearable computer components. Wearable computers must be portable, hands-free, context-aware, attention-getting (such as a ringing cell phone), and always on.[1] Moreover, they must have sensing technology that lets the computer be used without detailed instructions and respond to certain situations automatically.[2]

Researchers have used many kinds of digital sensors, such as motion, vision, and physiological sensors, to give wearable computers easy-to-use context awareness.[3,4] This article focuses on a low-power wearable sensing system for a wide range of users. I chose a head-mounted device because its location is close to where most perceptual activities, such as hearing and seeing, concentrate. To eliminate obtrusive wires between the headset and the other end of the computing system (and other computing components with wireless communication devices around the user), the system uses Bluetooth, a low-power, wireless data communication technology.

The architecture

The weight limitation of wearable-device equipment on each part of the user's body is extremely important; the wiring scheme might also affect the user's choice of clothes. Although wireless communication technology could potentially solve the latter problem, we encounter other issues concerning data reliability and security, inferior communication bandwidth, how each wireless device is powered, and so on.

Figure 1 shows the architecture of my proposed peripheral device, and Figure 2 illustrates each component, with quarters indicating rough dimensions. The headset's low weight (220 grams) and lack of rigid wires make this device acceptable for long time periods.

A Bluetooth radio module is placed on the left side of the user's head (see www.bluetooth.com for specifications). To reduce the headset's power consumption, the module's transmission power is set to approximately 1 mW (a class-two device in the Bluetooth specification). Because this transmission power is much less than that of cellular phones (in the order of 10 mW to 1 W), harmful microwave irradiation into the user's head is reduced.

On the same side as the Bluetooth module, a low-power eight-bit RISC microcontroller acts as a system controller; it includes a Bluetooth protocol stack and a bidirectional voice interface. The Bluetooth specification defines a built-in data encryption scheme, ensuring privacy of wireless data communication.

Another eight-bit RISC microcontroller with an up to eight-channel analog-to-digital converter per-

1094-7167/01/$10.00 © 2001 IEEE

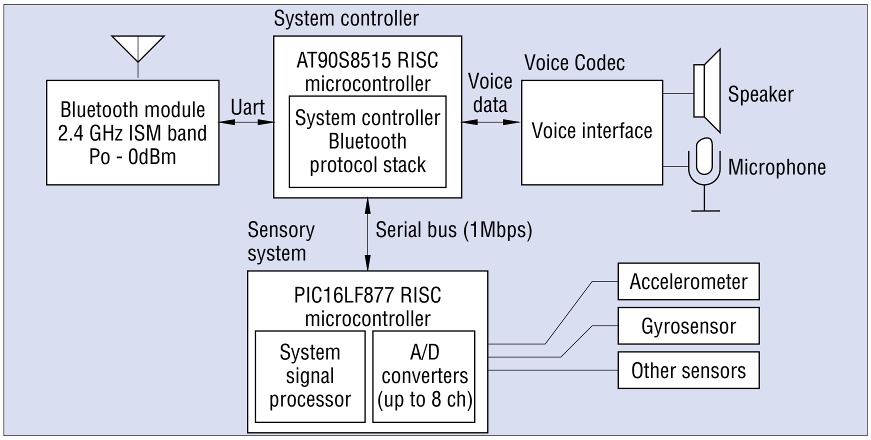


Figure 1. The architecture of a head-mounted peripheral device.

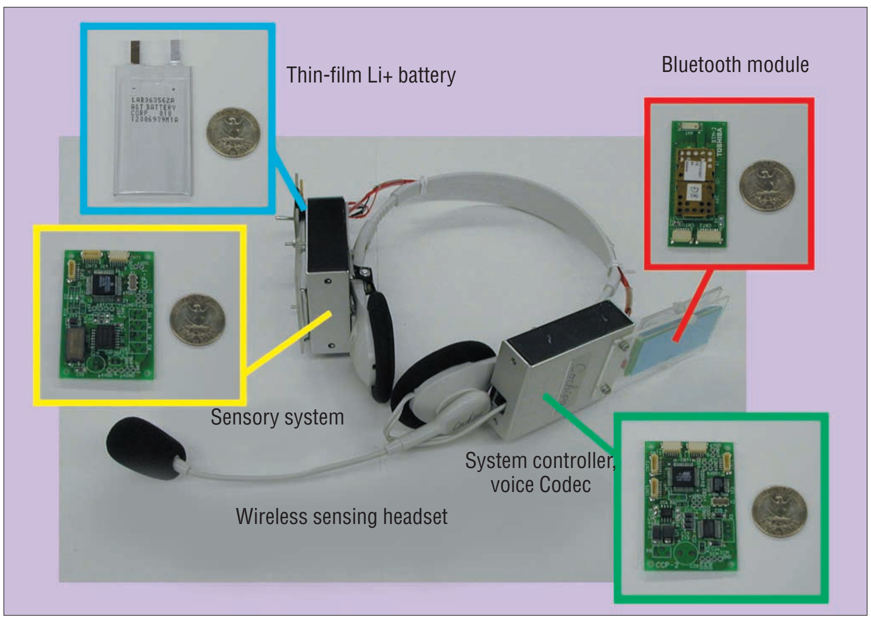


Figure 2. The wireless headset's components.

forms sensor signal processing. It has context (in other words, motion) recognition on the right side of the user's head, including a two-axis accelerometer and a single-axis ceramic gyrosensor (see Figure 3) near the user's right ear. Traveling through a high-speed serial data bus (1 Mbps) between the two RISC microcontrollers, messages consist of raw sensor signals, and recognized context symbols are transferred to the system controller. After producing the data packets for the Bluetooth radio module by mixing the voice data and messages from the sensory system, the headset transmits them to the other end of the system. Apparatuses on the other end could include a cellular phone, a personal digital assistant, a notebook PC, a fixed information service station on the street, and so forth. The computing system can be dynamically reconfigurable while enjoying the Bluetooth link's ad hoc networking capability. A high-performance microprocessor does not have to be the heart of the computing system.

Figure 4 shows each component's power consumption in the fully activated mode. In a worst-case scenario, the headset consumes 224 mW (3.3 V and 68 mA). To supply enough power for long time periods without adding extra weight, the system uses a thin lithium-ion film battery (3.8 V, 650 mAh, 15 g, and 3.8-mm thick) along with 3.3-V voltage regulators. The battery life extends to more than eight hours.

Sensors on the head-mounted device

As Figure 1 shows, the proposed device can use many kinds of sensors. Among them, I implemented two motion sensors because they use less power than vision sensors or voice-recognition systems. Figure 5 shows the arrangement of the sensors and definitions for each measurement, and Table 1 summarizes the sensors' specifications.

A shock perpendicular to the ground induced by a user motion such as walking is detected by the RISC microcontroller with the two-axis accelerometer. From the acceleration signal Ax, we can also measure the attitude of the user's head toward the ground in the form of a static acceleration due to the gravity and can measure the acceleration due to the user's backward and forward motion. Unlike the accelerometers, the gyrosensor detects the user's rotational motion and is not directly sensitive to a shock.

The combination of the accelerometer and the gyrosensor lets the device estimate the user's motion more precisely. Figure 6 shows the waveforms derived from the sensors while the user walks or looks around when standing still. Considering the sensors' bandwidths, I set the data-sampling period to 10 ms. During walking, periodic acceleration patterns due to footsteps show up in Az (see Figure 6). The signals from the gyrosensor also show periodic patterns corresponding to the rotational motion of the user's head while walking.

When the user stops, Az shrinks; the gyrosensor reports fairly large signals when the user looks around to find something. In this case, the maximum rotational angle from the center is approximately 30 degrees to the

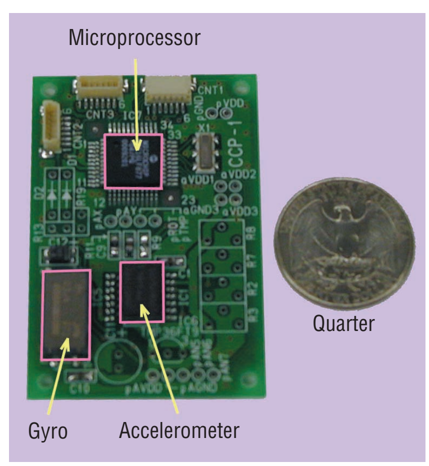


Figure 3. The sensory system board.

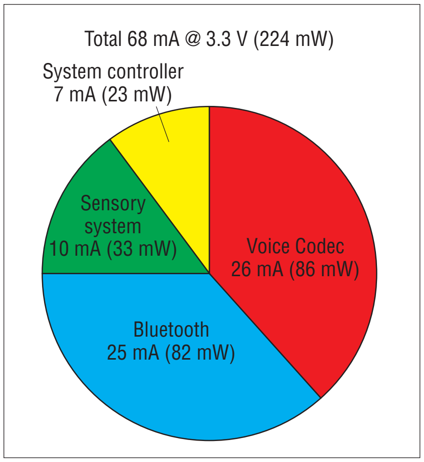


Figure 4. Power-consumption distribution.

Table 1. Specifications of motion sensors.

	Accelerometer	Gyrosensor
Bandwidth	100 Hz (–3 dB)	50 Hz (–3 dB)
Sensitivity	0.65 V/G	4.0 mV/(deg./sec.)
Measurement range	± 2.5 G	± 300 deg./sec.
Power consumption (including filter and amplification)	7 mW	20 mW

left and right. Although the rotational motion, which we can see in *Rv* in Figure 6, isn't fast (meaning signal intensity isn't high) when the user stands still and looks around, the *Rv* signal's intensity is sufficiently clear to distinguish the looking-around motion from the walking one. If the wearable device's obtrusiveness affects the user's motion, the obtained data will vary for different implementations, which makes designing context-recognition algorithms more difficult.

Onboard context recognition

As an example of sensory system design for human–computer interaction applications, I developed a real-time algorithm to determine how the user walks. To reduce power consumption, I designed the recognition process to be performed by a low-power RISC microcontroller in the headset. As mentioned earlier, this design also helps maintain robust recognition by eliminating unreliable wireless data links.

By investigating the motion of walking as well as the obtained signals from the sensors, I developed an efficient algorithm to derive the number of footsteps from the acceleration measurement. First, the RISC microcontroller detects acceleration exceeding a certain threshold value. In this case, it is preferable to detect an acceleration that has the same direction as gravity because the waveform's shape is usually sharper than that of the opposite polarity (see Figure 6). Additionally, if the user does not move at all, the maximum value of acceleration due to the gravity is –1 G. That is, changes in posture without a significant shock do not affect recognition.

Second, the microcontroller waits for a certain predetermined time (a dead time) to avoid spurious responses due to the complicated waveform around the place where the acceleration exceeds the predetermined threshold. In the experiment, 200 ms of waiting time provided sufficient accuracy even if the user ran quickly. I set the sampling period to 10 ms, which was enough to detect the specific waveform motion and the microcontroller's processing time. (Because the microcontoller has a throughput of approximately 1 million instructions per second, the system can perform nearly 10,000 instructions in the sampling clock period.)

Because the Bluetooth technology uses a half-duplex link with each slot length of at least 0.625 ms (due to the frequency-hopping scheme), the data can be delayed for 1.25 ms—even if the data link is error-free and has no additional delay. If noise causes a packet loss, the delay can become much larger. If the sampled data is transmitted to the other end, where signal processing occurs, the delay of the data packet transfer might decrease the real-time context recognition's performance. In other words, the sensory system works as a reflex mechanism of the nervous system—rapid and light procedures are done in local activities having a quick response time instead of a high processing power, leaving the higher information processing to other computing elements that the user might wear.

In addition to the number of footsteps, the sensory system can guess other kinds of

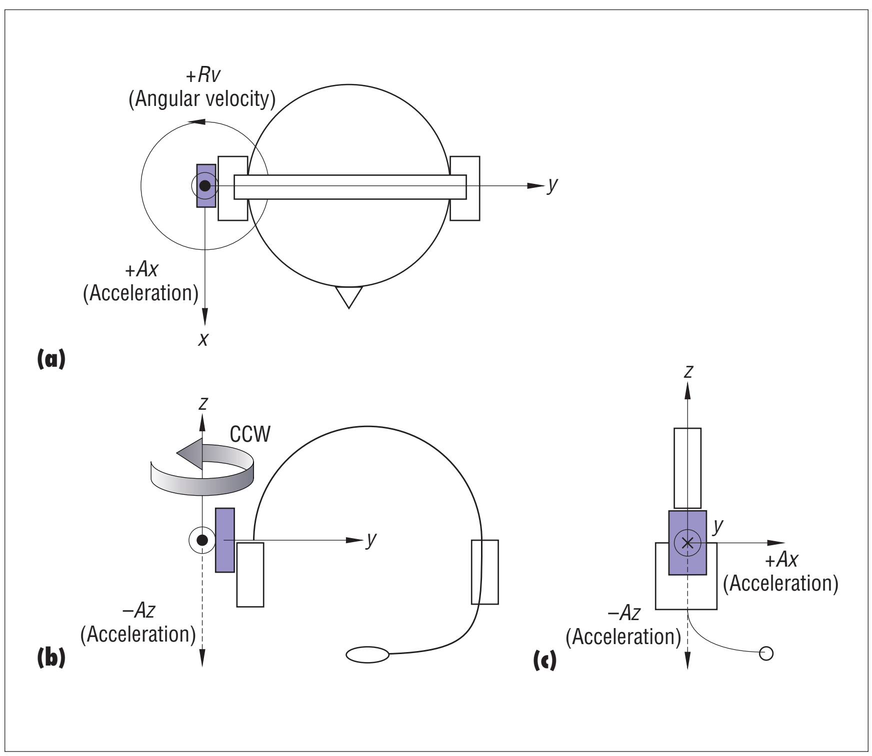


Figure 5. Definitions of motion measurements: (a) top view, (b) front view, and (c) side view.

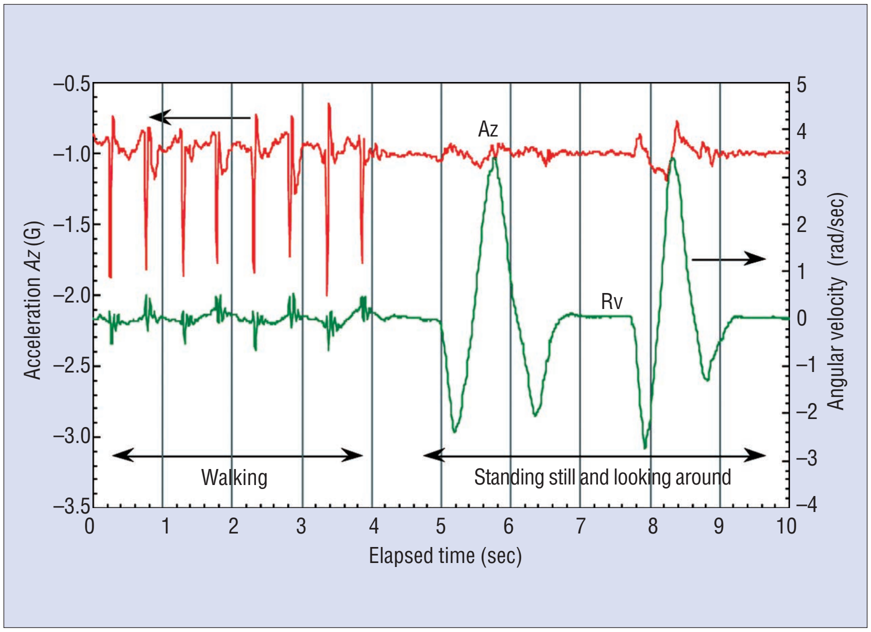


Figure 6. Waveforms from motion sensors.

motion. For example, the recognition system can determine if the user is nodding by having the microcontroller observe sharp acceleration waveforms in *Ax* from the sensor. In this case, because the attitude of the user's head changes and might produce fairly large acceleration signals, both the changing speed and intensity of acceleration should be compared to predetermined thresholds. By monitoring the angular velocity *Rv*, we can find the rotational motion of the user's head. So far, the headset can produce four context symbols in the form of four-bit digital data, where each bit indicates the detection of footsteps, rotating to the right, rotating to the left, and nodding, with some parameters such as each motion's intensity. Although there aren't many context symbols, they are both useful for modifying the functionality of wearable computing and to give the user power of expression.

Information about how the user moves can help annotate the data from other sensory systems such as video cameras or microphones. Although discriminating optical flows due to user motion consumes more processing power, the context symbols reduce power consumption by telling the system only about the user's motion. In fact, the gyrosensor on the headset was originally developed for handheld video cameras to remove image jitter due to vibration. By using footstep recognition, the system can signal when image processing should start comparing two consecutive frames.

Because computational power and other resources are limited in a wearable environment, selecting priorities for individual contexts is important. It would be hard for the proposed headset to guess the user's behavior precisely, because the user's motion alone usually contains insufficient information about it. Using other sensors to increase the guessing accuracy is one of the ways to solve the problem. On the other hand, it might be feasible to give the wearable system affordable functionalities of context-awareness under some predetermined scenarios. Pedestrian navigation is one of the most promising applications for the headset. If the user is standing still and looking around, the headset can guess that he or she is looking for something. Roughly speaking, there might be two possibilities in this case. First, the user might be lost. Second, the user might encounter something that makes him or her stop—for example, a red traffic signal or something attractive. In any case, asking the user is one of the best ways to let the wearable system "know" what is happening. Thanks to the proposed headset's nature—which has a voice interface—we can realize a dialogue-based computer–human interaction system by adding voice recognition and synthesizing components to the wearable system.

Figure 7 shows an application using both video and motion sensors. If the user is interested in the signboard, he or she gets near it and then stops. Detecting this kind of motion, a computer vision system using a wearable camera starts to recognize what is in front of the user. In this case, an invisible tagging system can make that computing much more useful.[5] The wearable system can recognize that the user is looking at the signboard by determining the signal from the infrared tag related to the signboard. After the registration process, further data communication can take place between the wearable system and the signboard with the Bluetooth link. Before starting time-consuming tasks such as downloading large amounts of data from the signboard, the system should ask if that's what the user wants to do.

Applications for context-aware headsets

In addition to interpreting user motions into a corresponding symbolic code, this

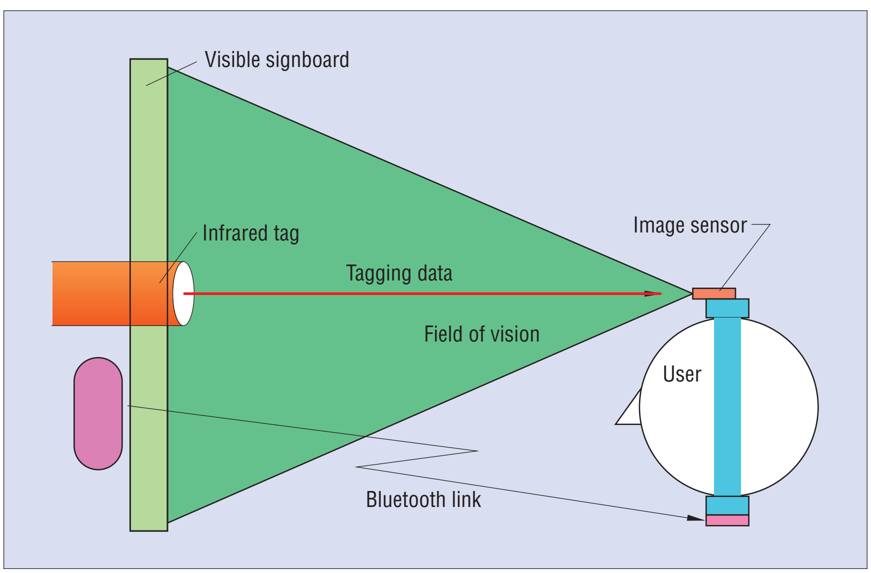


Figure 7. Infrared tagging system.

Figure 8. Communication enhancement.

device can also convert a motion into a corresponding sound (see Figure 8). The RISC microcontroller in the headset converts the user’s reaction to conversation into user-specific sound while determining the context symbols. Nonvolatile-memory devices in the headset can store the user-specific sounds corresponding to various context symbols. Therefore, the sounds can describe the context as well as the user’s identity.

The device can also interact with robotic devices. I experimented with it on a rover robot (made from Lego Mindstorms; see www.lego.com) using a Bluetooth link in the form of context symbols. Responding to my nodding motion, the rover robot shot a plastic bullet with an overhand throw. Because such a rover robot system can be controlled without computer-specific interfaces such as keyboards, pointing devices, or even voice commands, anyone can interact with it freely. Appropriate settings of context-recognition parameters can make the headset system acceptable for physically challenged users. That is to say, regardless of the type of communication, this headset can add a nonverbal communication path.

The combination of a minimized headset computer and a cell phone’s functionality could create promising technological avenues. Users who are far apart from each other could enjoy the expanded communication path in Figure 8. The bandwidth required to transfer context symbols is far narrower than that for power-consuming videos, which can achieve similar functionality. Because of its low-power, lightweight, unobtrusive capability for ad hoc networking and dynamic configuration, this proposed headset can be used as always-ready digital equipment for a wide range of information services (such as pedestrian navigation in mobile environments). In this sense, minimized wearable computers could become the most commonly used PCs on the mobile market.

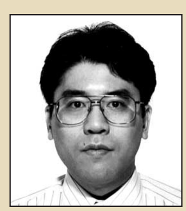
In the future, more effective context-aware implementations and attractive applications will be crucial for wearable computers to gain dominance in the PC market. ■

References

1. B.J. Rhodes, “The Wearable Remembrance Agent: A System for Augmented Memory,” *Proc. 1st Int’l Symp. Wearable Computers* (ISWC ’97), IEEE Computer Soc. Press, Los Alamitos, Calif., 1997, pp.123–128.
2. A. Pentland, “Looking at People: Sensing for Ubiquitous and Wearable Computing,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, Jan. 2000, pp.107–119.
3. J. Farringdon et al., “Wearable Sensor Badge and Sensor Jacket for Context Awareness,” *Proc. 3rd Int’l Symp. Wearable Computers* (ISWC ’99), IEEE Computer Soc. Press, Los Alamitos, Calif., 1999, pp. 107–113.
4. J. Healey and R.W. Picard, “StartleCam: A Cybernetic Wearable Camera,” *Proc. 2nd Int’l Symp. Wearable Computers* (ISWC ’98), IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp.42–49.
5. H. Aoki and S. Matsushita, “Balloon Tag: (In)visible Marker Which Tells Who’s Who,” *Proc. 4th Int’l Symp. Wearable Computers* (ISWC 2000), IEEE Computer Soc. Press, Los Alamitos, Calif., 2000, pp. 181–182.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Author



Soichiro Matsushita is a lecturer in the Department of Computational Science and Engineering at Toyo University. He is also a member of the Bio-Nano Electronics Research Center at the university. His research interests include the design of solid-state integrated circuits, bionano electronics, and wearable computing systems. He received his PhD in electronic engineering from the University of Tokyo. He is a member of the IEEE Computer Society. Contact him at gmatsu@acm.org.

Vision-Based Single-Stroke Character Recognition for Wearable Computing

Ömer Faruk Özer, *Bilkent University*
Oğuz Özün, C. Öncel Tüzel, and Volkan Atalay, *Middle East Technical University*
A. Enis Çetin, *Sabanci University and Bilkent University*

People want increasingly flexible and mobile ways to communicate using computers. Wearable computing offers advantages but making data entry easy remains a challenge. The authors discuss a new approach for data entry using a head-mounted digital camera to record characters drawn by hand gestures or by a pointer.

Particularly when compared to traditional tools such as a keyboard or mouse, wearable computing data entry tools offer increased mobility and flexibility. Such tools include touch screens, hand gesture and facial expression recognition, speech recognition, and key systems.

However, making data entry easy poses a challenge. New approaches (see the sidebar, "Useful URLs") such as one-handed chording keyboards help us understand the problems and complexities. Using the character recognition systems developed in document analysis, computer vision-based man–machine communication systems are possible.[1,2] For example, personal digital assistants let users write rather than type on a small keyboard, thanks to the success of unistroke, isolated character recognition systems.[3,4] In most of the new data entry approaches, the rate of data entry is lower than that of the traditional keyboard- or mouse-based entry. On the other hand, fast data entry systems require a learning phase most people would rather avoid.

In this article, we describe a new approach for recognizing characters drawn by hand gestures or by a pointer on a user's forearm captured by a digital camera. We draw each character as a single, isolated stroke using a Graffiti-like alphabet. Our algorithm enables effective and quick character recognition. The resulting character recognition system has potential for application in mobile communication and computing devices such as phones, laptop computers, handheld computers, and personal data assistants.

The recognition system and our algorithm

Consider this scenario: A user draws unistroke, isolated characters with a laser pointer or a stylus on their forearm or a table. A camera on their forehead records the drawn characters and captures each character in sequence. The image sequence starts when the user turns the pointer on and ends when they turn

Useful URLs

- **The septambic keyer**, http://wearcam.org/septambic
- **The Twiddler**, www.handykey.com
- **The EyeTap**, http://eyetap.org
- **The Pendragon project**, www.cc.gatech.edu/fce/pendragon
- **Multimodal conversational interaction**, http://vislab.cs.wright.edu
- **Visual gesture research**, www.ifp.uiuc.edu/~jy-lin/gesture/gesture.htm
- **User system ergonomics research**, www.almaden.ibm.com/cs/user.html

1094-7167/01/$10.00 © 2001 IEEE

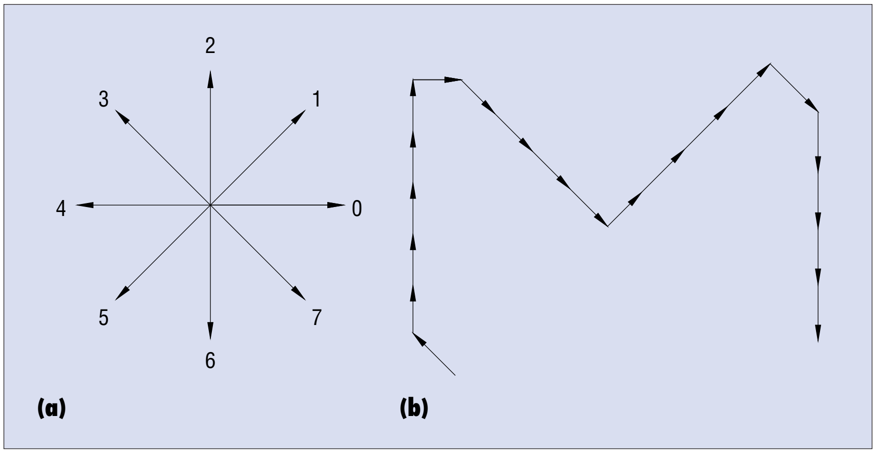


Figure 1. (a) Chain code values for the angles; (b) a sample chain-coded representation of the character *M* = 32222207777111176666.

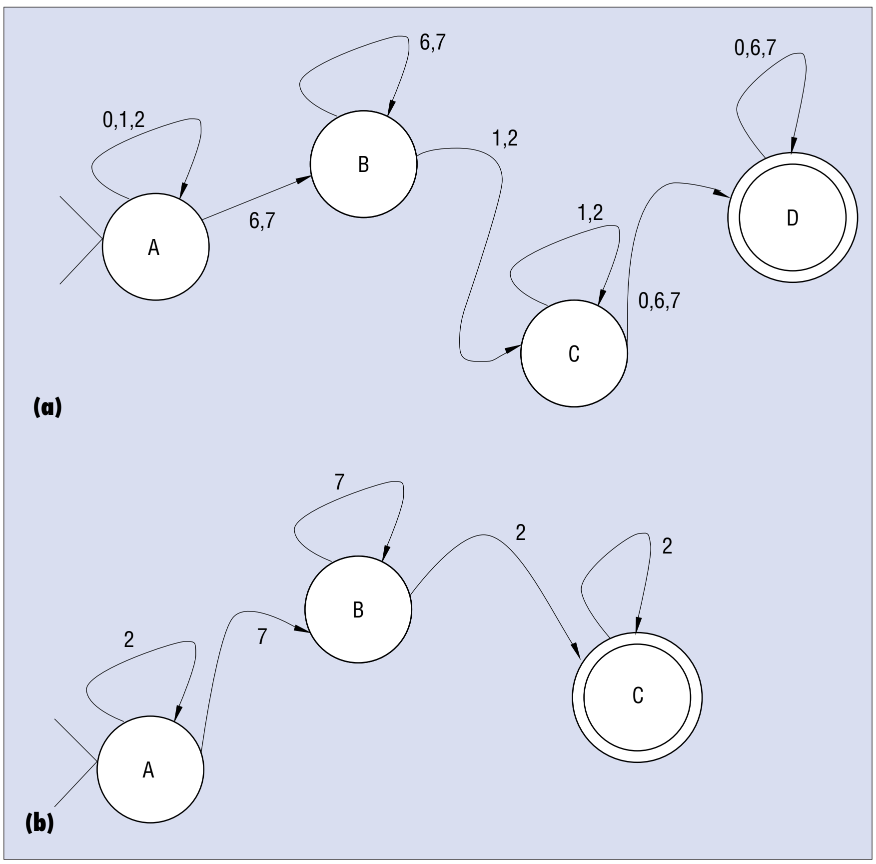


Figure 2. Finite state machines for the characters (a) *M* and (b) *N*.

it off. Thus, discontinuous pointer movements separate each character.

In our approach, a *chain code* describes the unistroke characters drawn. A chain code is a sequence of numbers between 0 and 7 obtained from the quantized angle of the laser point's beam in an equally timed manner. We extract chain code from the beam's relative motion between consecutive images of the video. The chain code is the input for the recognition system.

The recognition system consists of finite state machines corresponding to individual characters. The FSMs generating the minimum error identify the recognized character. However, certain characters such as *Q* and *G* might be confused in a feature set comprising only the chain code. Therefore, the system also considers the beginning and end strokes. The weighted sum of the error from a finite state machine and the beginning and end point error determines the final error for a character in the recognition process.

Our algorithm for character recognition consists of four steps, described in the following paragraphs.

Step 1, extraction of chain code. The system

- finds the position of the red mark the laser pointer produces in each frame,
- generates a chain code according to the angle between two consecutive mark positions, and
- determines beginning and end point coordinates together with the coordinate of a rectangle enclosing the character.

Step 2, analysis using finite state machines. The system

- applies the chain code as input to each state machine,
- determines state changes (additionally, the system increases an error counter by one if a change is not possible according to the current FSM),
- eliminates the corresponding character if a chain code does not terminate in the final state, and
- adds up errors in each state to find the final error for each character.

Step 3, accounting for errors due to beginning and end points. The system

- normalizes beginning and end points of a stroke with respect to the enclosing rectangle,
- determines if the width or the height is larger than a given threshold (if so, it isn't considered a feature), and then
- calculates an error value from the comparison of the normalized beginning and end points of the input character and the candidate character stroke.

Step 4, determining characters. The system

- weights and adds state machine error and position error, and

- recognizes the character with the minimum error.

Figures 1 through 3 illustrate our algorithm. About 20 consecutive images are merged to obtain the *M* image shown in Figure 1b and 3d; the corresponding chain code representation is 32222207777111176666. The FSM for the character *M* is shown in Figure 2a. Consider the laser beam traces of four characters shown in Figure 3.

When the chain code is applied as an input to this machine, the first element, 3, generates an error and the error counter is set to 1. The second element of the chain, 2, is a correct value at the FSM's starting state so the error counter remains at 1 after processing the input 2. The FSM remains in the first state with the other 2s and also with the subsequent 0, as 0, 1, and 2 are the inputs of the machine's first state for *M*. Input 7 makes the FSM go to the next state, and the subsequent three 7s let the machine remain there. Whenever the input becomes 1, the FSM moves to the third state. The machine stays in this state until the single 7 input, and this makes FSM go to the final state. The rest of the input data, 6, makes the machine stay in the final state, and when the input is finished, the FSM terminates. For this input sequence, 1 is the machine's error for character *M*. In practice, this sample chain code determines all other characters using FSMs. However, the other FSMs generate either greater or infinite error values. You can easily see this on the character *N*'s FSM (see Figure 2b). If *M*'s chain code string is an input to this machine, it will never reach the final state and the error will be set to infinity.

Both the time and space complexity of the recognition algorithm are $O(n)$, where n is the number of elements in the chain code. The FSM recognition algorithm is robust as long as the user does not move his arm or the camera while writing a letter. Small changes due to hand trembling while writing can be corrected automatically by look-ahead tokens to improve the recognition rate. The look-ahead tokens act as a smoothing filter on the chain code. Instead of using deterministic FSMs, characters can also be modeled by hidden Markov models (stochastic FSMs) to further increase the system's robustness, but this also increases computational cost.

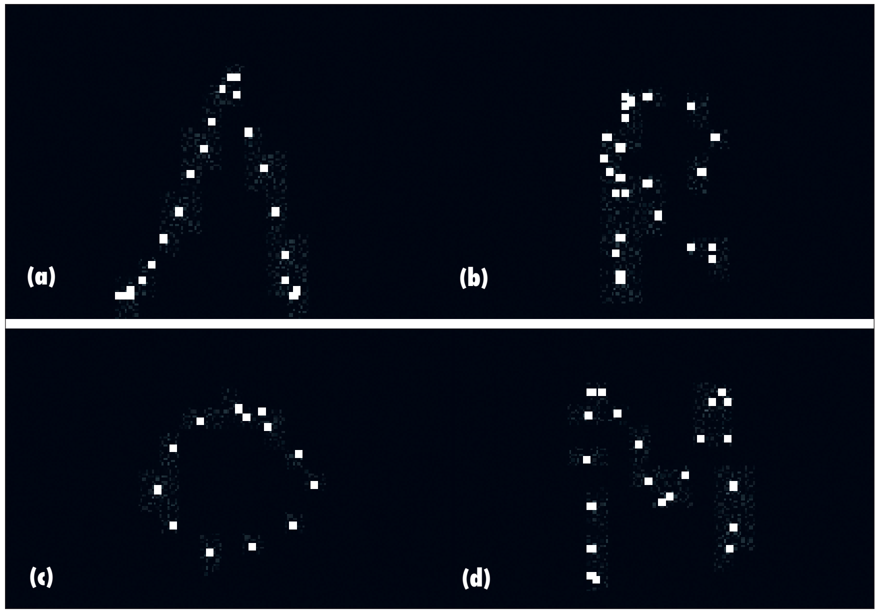


Figure 3. Laser beam traces generated by image sequences corresponding to (a) lambda (*A* in Graffiti), (b) *R*, (c) *O*, and (d) *M*.

Video processing

To extract chain code from the video, marker positions for the images corresponding to a character are processed. If the marker is in the initial frame, you can track it in the consecutive images. In our experiments, we used a red laser pointer to write the characters. Then, we decomposed the images into red, green, and blue components. *Thresholding*—a simple image-processing operation—followed by a connected component analysis identifies the red mark. If you use hand gestures, you might need a skin filter. We can similarly extract and trace other pointers (for example, a pen tip).

A laser pointer is the most robust text entry device in changing lighting and background conditions. As discussed earlier, in an image sequence corresponding to a word, discontinuous pointer movements separate characters. For a laser pointer, at the end of each character the user turns off the light. This marks the end of each character. For each character, we segment the video based on the jumps of the laser pointer's red mark. While the user is writing a character, the transition of the pointer positions in consecutive images should be smooth because the user writes only unistroke characters. The subsequent character will start at a relatively different position because the characters are written one at a time. Therefore, using a laser pointer naturally creates a deliberate discontinuity between two characters.

Two problems mainly arise during image capture and processing: distortion due to perspective projection and marker occlusion. Character distortion occurs when the user draws the hand gestures in a nonorthographic manner. Perspective distortion up to about 45 degrees of difference defined by the laser pointer (or regular pointer) between the camera and the forearm's tangent plane does not affect character recognition. The system fails after 45 degrees because the chain code used in character representation has a quantization level of 45 degrees (the unit circle is represented by eight directions). You can overcome this problem by either increasing the quantization levels and modifying the FSM models accordingly or by using Steve Mann's projective geometry methods[5–7] to provide an efficient solution with the help of feedback from a viewfinder. We don't consider occlusion in this system, because we assume the camera captures the images in front of the marker.

Experimental results

We used a red laser pointer, black background fabric, and a Web camera (an ordinary Philips PC Camera with a Tekram VideoCap C210 capturing card) in our experiment. The Web camera produces 160×120 pixel color images at 13.3 frames per second. We used an Intel Celeron 600 processor with 64 Mbytes of memory for all processing.

We have not yet implemented our system on a wearable computer; however, we believe our experimental setup and algorithm illustrate the results we would find with wearable computing applications. The processor we used performs similarly to the processors mentioned in current wearable computers. Furthermore, the Web camera used in our experiments has very similar characteristics with the head-mounted cameras used in wearable computers or the EyeTap (http://eyetap.org).

In our experiments, the user draws a Graffiti-like character using the red pointer on dark background material. In other unistroke recognition systems, you can achieve very high recognition rates.[4] In our system, in spite of perspective distortion, you can attain a recognition rate of 97 percent at a writing speed of about 10 words per minute. We also noted that the recognition process is writer-independent and writers required little training. We used the Graffiti-like alphabet because it resembles the Latin alphabet, and most people can use it without extra effort. Users can also define other single-stroke characters to use as bookmarks or pointers to databases, for example. Although it might be easy to learn other text entry systems, some people are reluctant to take the time to learn unconventional text entry systems. Computationally efficient, low power consuming algorithms exist for the recognition of unistroke characters. We can implement these algorithms in real time with very high recognition accuracy. After a user studies the Graffiti-like alphabet for a few minutes, about 86-percent accuracy is possible. After some practice, accuracy improves to about 97 percent. Almost 100-percent accuracy seems possible.[8]

To estimate the above recognition rate, we used at least 50 samples for each character and a total of 1,354 characters. The system requires an average of 18 image frames per character. Typically, a user draws these in less than 1.5 seconds. This means a data entry rate of more than 40 characters per minute on average. Users can improve writing speed if they spend time learning better ways to write certain characters. For example, the characters *I* and *T* can be drawn and recognized with almost 100-percent accuracy using only three to four frames. In contrast, the character *B* needs at least 50 frames (or more than 3.35 seconds) for reasonable recognition rate accuracy. Perspective distortion also plays a minor role in the system because everything is two-dimensional. In our experiments, we observed that degradation in recognition is, at most, 10 percent around a 45-degree difference between the writing plane and the camera.

We also conducted several tests under different lighting conditions. In daylight, the background's pixel value is about 50 whereas the pixel value of the laser pointer's beam is about 240. In incandescent light, the background's pixel value is about 180 whereas the beam's pixel value is about 250. In fluorescent light, the background's pixel value is about 100 whereas the beam's pixel value is about 240. In all cases, we can easily identify the laser pointer's beam against the dark background because enough contrast exists, especially if the user also wears a dark, solid color. If the user writes characters with her finger, we expect a slightly lower recognition rate. Writing with a finger is much more convenient than writing with a laser pointer; however, detecting the laser pointer's beam is simpler for image analysis.

Our current system's overall writing speed is below the 20-wpm composition rate reported for Graffiti on a PDA.[8] This is because a wearable camera's frame rate is much smaller than a PDA touchscreen's sampling rate. However, a PDA requires much slower writing movements when compared to our approach. Our recognition algorithm is also more complex and robust than the simple recognition algorithms used in PDAs.

Our system's writing speed is also lower than the 35- to 40-wpm transcription speeds of the septambic keyer and the Twiddler. However, regardless of the keyboard, composition writing speed is below 20-wpm for most people. We believe that in a wearable computing environment the composition speed rather than the transcription speed is important. Furthermore, we can achieve the 20 wpm writing speed with very high accuracy in our system (or in today's wearable computing technology) if we use an optimized unistroke alphabet[4] instead of a Graffiti-like alphabet. In such a case, the user would have to learn an alphabet consisting of even more simple strokes.

While our approach hasn't been implemented in wearable computing yet, several interesting applications are possible. For example, our current system is well suited for taking notes while watching a presentation if the camera has a viewfinder.[9–11] The viewfinder provides a feedback loop so the user can review and correct any errors in pointer-written characters as they occur.

We are working on generalizing the system to recognize continuous writing with a finger or stylus. We are also studying an alternative way to recognize characters using a wearable keyboard image and a laser light. You enter characters by shining light onto the character's location on the keyboard image. A finger or stylus can be used to mask the key locations to enter text. If you use an optimized keyboard image (such as the Pendragon Project's Cirrin or IBM's Metropolis), text entry speed can exceed the ordinary keyboard. ■

Acknowledgments

The authors thank the *IEEE Intelligent Systems* guest editor, Steve Mann of the University of Toronto, and the anonymous reviewers whose comments and corrections significantly improved this article.

References

1. O.N. Gerek et al., "Subband Domain Coding of Binary Textual Images for Document Archiving," *IEEE Trans. Image Processing*, vol. 8, no. 10, Oct. 1999, pp. 1438–1446.
2. E. Oztop et al., "Repulsive Attractive Network for Baseline Extraction on Document Images," *Signal Processing*, vol. 75, no. 1, Jan. 1999, pp. 1–10.
3. D. Goldberg and C. Richardson, "Touch-Typing with a Stylus," *Proc. '93 Conf. Human Factors in Computing Systems*, ACM Press, New York, 1993, pp. 80–87.
4. I.S. MacKenzie and S. Zhang, "The Immediate Usability of Graffiti," *Proc. Graphics Interface '97*, Morgan Kaufmann, San Francisco, 1997, pp. 129–137.
5. S. Mann, "Further Developments on Head-Cam: Joint Estimation of Camera Rotation + Gain Group of Transformations for Wearable Bi-Foveated Cameras," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 4, IEEE Press, Piscataway, N.J., 1997, pp. 2909–2912.
6. S. Mann and R.W. Picard, "Video Orbits of the Projective Group: A Simple Approach to Featureless Estimation of Parameters," *IEEE Trans. Image Processing*, vol. 6, no. 9, Sept. 1997, pp. 1281–1295.
7. S. Mann, "Humanistic Computing: WearComp as a New Framework and Application for Intelligent Signal Processing," *Proc. IEEE*, vol. 86, no.11, Nov. 1998, pp. 2123–2151.
8. S. Zhai, M. Hunter, and B.A. Smith, "The Metropolis Keyboard: An Exploration of Quantitative Techniques for Graphical Keyboard

Design," *Proc. 13th ACM Symp. User Interface Software and Technology (UIST 2000)*, ACM Press, New York, 2000, pp. 119–128.

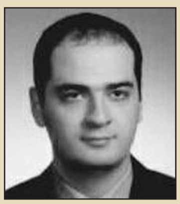
9. S. Mann, "WearCam (The Wearable Camera): Personal Imaging Systems for Long-Term Use in Wearable Tetherless Computer-Mediated Reality and Personal Photo/ Videographic Memory Prosthesis," *Proc. 2nd Int'l Symp. Wearable Computers*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 124–131.

10. S. Mann, "Smart Clothing: the Wearable Computer and WearCam," *Personal Technologies*, vol. 1, no.1, Mar. 1997.

11. S. Mann, "Telepointer: Hands-Free Completely Self-Contained Wearable Visual Augmented Reality without Headwear and without Any Infrastructural Reliance," *Proc. 4th Int'l Symp. Wearable Computers*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 177–178.

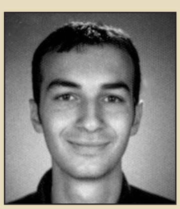
For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Authors



Ömer Faruk Özer received a BSc in electrical and electronics engineering from Bilkent University. Contact him at the Department of Electrical Engineering, Bilkent University, Ankara TR-06533, Turkey; oozer@ug.bcc.bilkent.edu.tr.

Oğuz Özün is a research assistant at Middle East Technical University. His research interests include computer graphics and computer vision, particularly three-dimensional model reconstruction from a sequence of images. He received a BSc in computer engineering from Middle East Technical University. Contact him at the Dept. of Computer Eng., Middle East Technical Univ., Ankara TR-06531, Turkey; oguz@ceng.metu.edu.tr.



C. Öncel Tüzel works for Meteksan Sistem at Bilknet University. His research interests include computer graphics and computer vision, particularly image-based rendering. He received a BSc in computer engineering from Middle East Technical University. Contact him at Meteksan Sistem ve Bilgisayar Teknolojileri, Bilkent University, Beytepe Koyu No:5, Bilkent, Ankara TR-06533, Turkey; oncel.tuzel@meteksan.com.tr.

Volkan Atalay is a faculty member at Middle East Technical University. Previously, he was a visiting scholar at the New Jersey Institute of Technology. His research interests include computer vision and document analysis. He received a BSc and an MSc in electrical engineering from Middle East Technical University and a PhD in computer science from the Université René Descartes-Paris V, Paris, France. He is a member of the IEEE Computer Society and the Turkish Pattern Recognition and Image Analysis Society. Contact him at the Dept. of Computer Eng., Middle East Technical Univ., Ankara TR-06531, Turkey; volkan@ceng. metu.edu.tr.

A. Enis Çetin is a faculty member of Sabanci University in Istanbul and is also currently on leave from Bilkent University. Previously, he was an assistant professor of electrical engineering at the University of Toronto, Canada, and a visiting professor at the University of Minnesota. He received a BSc in electrical engineering from Middle East Technical University and an MSE, and a PhD from the Moore School of Electrical Engineering, University of Pennsylvania. He is an associate editor of *IEEE Transactions on Image Processing* and a member of the DSP technical committee of the IEEE Circuits and Systems Society. Contact him at the Faculty of Eng. and Natural Sciences, Sabanci Univ., Orhanli TR-81474 Tuzla/Istanbul, Turkey; cetin@sabanciuniv.edu.

Toward Context-Aware Computing: Experiences and Lessons

Joshua Anhalt, Asim Smailagic, Daniel P. Siewiorek, and Francine Gemperle,
Carnegie Mellon University
Daniel Salber, Sam Weber, Jim Beck, and Jim Jennings, *IBM T.J. Watson Research Center*

To minimize distractions, a pervasive-computing environment must be context aware. The authors define an activity–attention framework for context-aware computing, discuss the spatial and temporal aspects of applications they developed, and introduce a pervasive-computing architecture.

The effects of Moore's law are apparent everywhere: Chip density, processor speed, memory cost, disk capacity, and network bandwidth are improving relentlessly. As computing costs plummet, a resource that we have ignored until now becomes the limiting factor in computer systems—user attention, namely a person's ability to focus on his or her primary task.

Distractions occur especially in mobile environments, because walking, driving, or other real-world interactions often preoccupy the user. A pervasive-computing environment that minimizes distraction must be context aware, and a pervasive-computing system must know the user's state to accommodate his or her needs.

Context-aware applications provide at least two fundamental services: spatial awareness and temporal awareness. Spatially aware applications consider a user's relative and absolute position and orientation. Temporally aware applications consider the time schedules of public and private events. With an interdisciplinary class of Carnegie Mellon University (CMU) students, we developed and implemented a context-aware, pervasive-computing environment that minimizes distraction and facilitates collaborative design.

Our approach

To identify the types of distraction that occur during the design process, we created an activity–attention matrix—the Distraction Matrix (see Figure 1). The Distraction Matrix categorizes activities as information (active and passive), communication (artificial, formal, and informal), and creation (contribution). Subcategories specify the types of primary activity within each category. For example, receiving information is a type of active-information activity, and initiating communication is a type of artificial-communication activity.

We based each distraction's location on how long it interrupts a primary activity. We categorized interruption durations as *snap*, *pause*, *tangent*, and *extended*. A snap distraction is one you usually complete in a few seconds, such as checking your watch; it should not interrupt your primary activity. A pause distraction involves stopping the primary activity, switching to a related one, and then switching back within a few minutes. Pulling over to the side of the road and checking directions is an example. A tangent distraction, such as receiving an unrelated phone call, is of medium duration and is unrelated to your primary activity. An extended distraction, such as stopping at a motel and resting for the night, is a relatively long-term interruption of your primary activity.

Applications

We equipped the campus with 400 wireless-networking access points, enabling wireless coverage for the entire campus. To move distractions toward the Distraction Matrix's left (snap) side, we implemented a complementary set of interactive applications and services that support mobile team-design activities. (See the related sidebar for information on relevant work in context-aware computing.)

Portable Help Desk. Because they have many meetings at various times and locations, students are often

1094-7167/01/$10.00 © 2001 IEEE

Time →

		Snap	Pause	Tangent	Extended
Information					
Active	Receiving	Message arrival			Audio, Walkman
Active	Notifying	Information access			Transferring files from network
Active	Monitoring	Auction			
Active	Serendipity	Stocks, sports, matching similar needs Free food			Reading news
	Seeking	Line length Bus arrival Locate person	Exam calendar Software or hardware help Calendaring Navigation	Looking for class notes Who else is doing this now? Access personal data	
Passive	Browsing		Information on Web or built environment	Poster, bulletin board information	Web research
Passive	Finding				Reviewing class notes
Passive	Verifying		Recall previous queries Double-checking information		
Communication					
Artificial	Initiating	S.O.S. emergency	Introductions	Team building Collaborative work	Chatting (public or private)
Artificial	Participating	Instant messaging	Queries	Event planning Assassins game Social planning	
Informal	Broadcasting		Information exchange Scheduling	Posting information to bulletin board Advertising	
Formal		One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people
Creation					
Contribution	Recording	Remember this! Add a to-do or call list		Class note taking Meeting	Generating messages
Contribution	Synthesizing		Forwarding x to y	Filling out survey Registration	Summarizing lecture
Contribution	Generating			New ideas Adding information to existing projects	Mobile-tool building

Figure. 1. The Distraction Matrix. We based each distraction's location on the primary activity it interrupts and that interruption's duration (increasing from left to right).

unsure of where their next meeting is supposed to take place. The ability to observe team members' locations on campus helps students determine a meeting's location. The Portable Help Desk (PHD) application, a spatially aware system, confers that ability. It lets a user build maps of the immediate area, including colleague and static- and dynamic-resource locations, and quickly retrieve contact and resource availability information. While tracking a user's colleague, PHD displays that colleague's contact information.

The application can also display printer queues, restaurant hours, and stock of carbonated beverages and food in connected vending machines. Figure 2 shows the activities the PHD system supports and the attention each demands of the user.

We built both visual and audio interfaces for the PHD, each of which supports users in different contexts. The visual interface, designed for stationary use, is richer, but for a user who is walking around, the hands-free audio interface, Speech-PHD, is less distracting.

Figure 3 illustrates PHD's visual interface. The user selects people and resources in the left pane, and information about those people and resources appears in the middle pane. The right pane displays a campus map locating the selected people and resources.

Speech-PHD accesses the same database as the visual interface, so all responses are formatted similarly. Figure 4 is a transcript of the same queries that Figure 3 demonstrates. Because PHD knows the user's current location, it can answer questions such as "Where is the nearest ATM?"

PHD delivers information to the user in both proactive and user-driven manners. A user receives proactive information when engaging infrastructure resources such as printers. For instance, when the user begins a print job, PHD will alert him or her if a large print queue exists and suggest a nearby printer with a shorter queue. PHD can also suggest a printer near a destination to which a user is en route.

In terms of user-driven information, a design group waiting for a colleague can use PHD to locate the missing colleague and estimate his or her arrival time. The group also has access to the colleague's phone numbers. Essentially, PHD helps a group avoid repeating the beginning of a meeting for every late member. When the team members are getting hungry, they can look up the hours of nearby restaurants or check whether the soda machine is full.

Matchmaker. For large projects and design groups, no single individual has the expertise to perform every task. The Matchmaker application lets a user rapidly identify an expert user with the knowledge to help solve a problem. An expert's suitability depends on many factors, such as technical expertise, friendliness, proximity, and availability. Matchmaker infers expertise and skills by observing an expert's track record rather than by asking him or her explicitly. Matchmaker uses temporal context to determine an expert's availability and spatial context to determine the expert's distance from the user.

The Matchmaker system connects a user's query with an expert user who

- is nearby,
- is available,
- has a profile listing the skills needed, and
- has a history of answering similar questions.

Because the located expert is near the user initiating the question, he or she avoids wasting time moving to the user. After contacting the expert with the question, the Matchmaker system requests feedback from the expert to determine if he or she is best suited to answer the question. The database then updates its profile of the queried expert to increase expert-selection accuracy.

We have instantiated the Matchmaker system, letting users efficiently contact CMU's School of Computer Science Computing Support Group to resolve queries. The CSG maintains an extensive database of previously answered queries; this information lets the Matchmaker system generate profiles of CSG experts. Figure 5 shows some of the activities Matchmaker supports.

Figure 6 shows Matchmaker's system architecture. Matchmaker sends the user's query and the problem's location to the server. The server sends the query to the information-retrieval partition, which searches the database for similar queries, experts who answered those queries, and experts with similar knowledge. The central server sends the returned list of experts to the matchmaking partition, which compares the experts' locations and schedules

The Roots of Context-Aware Computing

Steve Mann introduced *humanistic intelligence*,[1,2] proposing it as a new signal-processing framework in which the processing apparatus supports and depends on the user's natural capabilities of body and mind. (For more on humanistic intelligence, see the Guest Editor's Introduction in this issue.) Anind K. Dey, Daniel Salber, Gregory D. Abowd, and Masayasu Futakawa designed a software architecture to let developers create context-aware applications.[3,4] Thad Starner, Bernt Schiele, and Alex Pentland developed context-aware user interfaces that use body-mounted, environment-looking cameras and machine-vision techniques.[5] Kristof Van Laerhoven and Ozan Cakmakci use body-mounted sensors to determine a user's activity and infer the user's context.[6] Gerd Kortuem, Zary Segall, and Martin Bauer describe a wearable computer that alters its user interface based on devices and services in the user's environment.[7]

References

1. S. Mann, "Humanistic Intelligence: 'WearComp' as a New Framework and Application for Intelligent Signal Processing," *Proc. IEEE*, vol. 86, no. 11, Nov. 1998, pp. 2123–2151.
2. S. Mann, "Humanistic Intelligence," *From the 1997 Ars Electronica Catalog*, www.wearcam.org/ars/hi.html (current 7 June 2001).
3. A.K. Dey, D. Salber, and G.D. Abowd, "Context-based Infrastructure for Smart Environments," *Proc. 1st Int'l Workshop on Managing Interactions in Smart Environments* (MANSE 99), Springer-Verlag, New York, 1999, pp. 114–128.
4. A.K. Dey et al., "An Architecture to Support Context-Aware Applications," tech. report GIT-GVU-99-23, Graphics, Visualization, and Usability Center, Georgia Tech, Atlanta, 1999.
5. T. Starner, B. Schiele, and A. Pentland, "Visual Contextual Awareness in Wearable Computing," *Proc. 2nd Int'l Symp. Wearable Computers* (ISWC 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 50–57.
6. K. Van Laerhoven and O. Cakmakci, "What Shall We Teach Our Pants?," *Proc. 4th Int'l Symp. Wearable Computers* (ISWC 00), IEEE CS Press, Los Alamitos, Calif., 2000, pp.77–83.
7. G. Kortuem, Z. Segall, and M. Bauer, "Context-Aware, Adaptive Wearable Computers as Remote Interfaces to 'Intelligent' Environments," *Proc. 2nd Int'l Symp. Wearable Computers* (ISWC 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 58–65.

Time →

		Snap	Pause	Tangent	Extended
Information					
Active	Receiving				
	Notifying				
	Monitoring	Fred is coming Print-queue alert			
	Serendipity				
	Seeking	Check print queue	Seek resources Find person		Access personal data
Passive	Browsing				
	Finding				
	Verifying				
Communication					
Informal Artificial	Initiating				
	Participating				
	Broadcasting				
Formal		One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people
Creation					
Contribution	Recording	Request event notification			
	Synthesizing	Select printer from suggested list			
	Generating				

Figure 2. Distraction Matrix for Portable Help Desk.

to the query's context. Then, the system notifies the chosen expert of the query.

Privacy Guard. PHD offers a valuable service to collaborating work groups, but its location-sensing ability is a liability. Therefore, we developed Privacy Guard to help users protect their information. Privacy Guard enables basic privacy policies and advanced expressions describing which users, groups, and time periods PHD can and cannot report. Figure 7 illustrates Privacy Guard's architecture.

The location-sensing service client derives the user's location from the wireless-network card and sends that location to the central server. Users update the central server with permissions. When the server receives a query for a user's location, it compares the client against the target's permissions. Accordingly, the server then sends the client the target's location or a refusal to answer the request.

Context-aware agents. Busy groups tend not to have abundant time to browse calendars, check for new email, or read bulletin boards. Therefore, we developed context-aware agents to deliver relevant information when a user needs that information. When the user is not engaged in more important activities, context-aware agents display appointments, urgent emails, and interesting calendar events. The agents are proactive: they monitor public and private calendars and email accounts and deliver information to the user instead of requiring the user to poll the relevant sources. Their goal is to provide intelligent calendar management, including setting schedules and resolving conflicts with other users' calendars while accounting for the location and available resources for a meeting. We have implemented the following three agents:

- *Notification Agent* alerts a user who passes within a certain distance of a location that a task on his or her to-do list identifies. For example, if a user is near his or her mailbox, the agent alerts the user if a package is waiting.
- *Meeting-Reminder Agent* alerts a user who is likely to miss a meeting. The system identifies the time the meeting will start and determines the travel time there from the user's current location.
- *Activity-Recommendation Agent* recommends activities and meetings, based on the user's interests, that the user might like to attend. For example, consider a user who sets his Activity-Recommendation Agent to inform him when free food is available. As the user walks through a building, the system identifies a meeting with free food upstairs and notifies the user.

Figure 8 shows an example user interface for the Activity-Recommendation Agent.

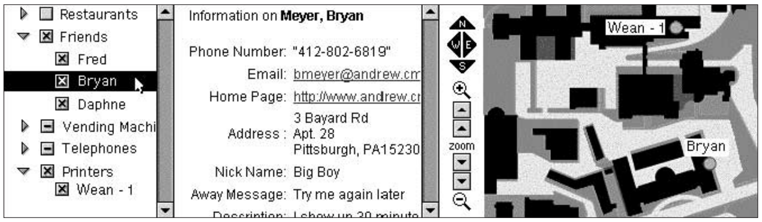


Figure 3. Portable Help Desk visual interface.

User:	"Locate Bryan."
Speech-PHD:	"Bryan is located in Hamburg Hall."
User:	"What is Bryan's phone number?"
Speech-PHD:	"Bryan's phone number is 412-802-6819."

Figure 4. Transcript from Portable Help Desk audio interface Speech-PHD.

The user defines his or her interests, enabling the agent to recommend upcoming activities. The agent categorizes interests by activity and keywords. The user can access the interface to find upcoming recommendations if he or she doesn't want to wait for the Activity-Recommendation Agent's notification.

We designed the context-aware agents to function as services that simpler applications can use.

The pervasive-computing environment

Mobile computing poses challenges such as intermittent and variable-bandwidth connectivity and client-resource constraints imposed by weight and size considerations. We based our pervasive-software architecture on one that the IBM T.J. Watson Research Center proposed. The class's prototype used Hewlett-Packard Jornada 680 palmtop computers and Itsy/Cue wearable computers communicating through Lucent Wavelan cards on Wireless Andrew.[1]

Our architecture's main goal is to let users seamlessly move work between devices. The architecture moves the application to a network-connected server, leaving only a minimal interface on the client device. Any device implementing the interface can then reattach the server running

Time →

		Snap	Pause	Tangent	Extended
Information					
Active	Receiving Notifying Monitoring Serendipity	Receive request for help Completion of task notification			
	Seeking		User searches through list of possible solutions returned by system Finding someone to help	User tries suggested solutions	
Passive	Browsing Finding Verifying				
Communication					
Artificial	Initiating Participating				Expert and user collaborate
Informal	Broadcasting				
Formal		One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people	One-to-one communication One-to-group communication One-to-all-possible communication broadcast to unknown people
Creation					
Contribution	Recording Synthesizing Generating		User initiates query		

Figure 5. Distraction Matrix for Matchmaker.

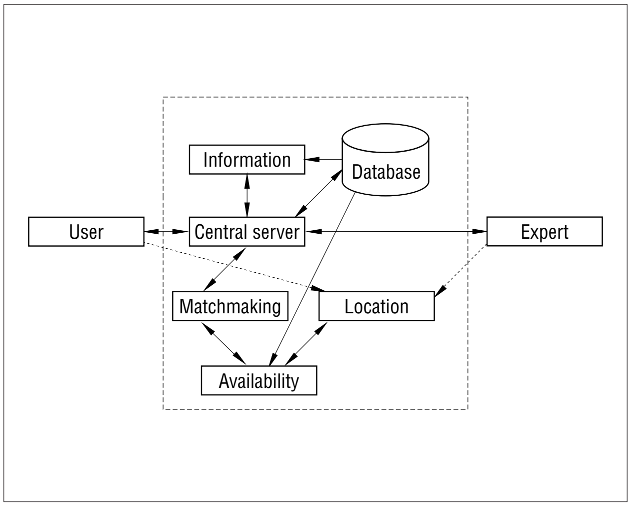


Figure 6. Matchmaker system architecture. The system receives the user's query and matches it to an appropriate expert user. It then locates the expert and notifies him or her of the query.

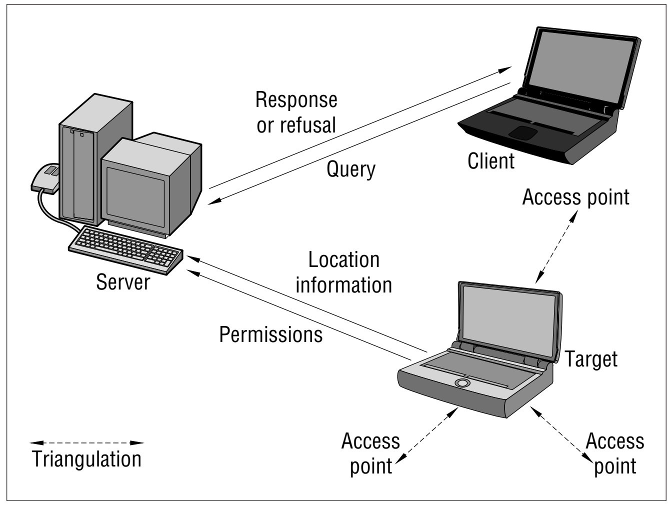


Figure 7. Privacy Guard architecture. By incorporating permissions, Privacy Guard limits a client user's access to a target user's contact and location information.

the application. To address mobile computing's connectivity issues, the architecture includes elements that preserve dataflow between the server and the devices. Optimizing data for device capabilities maximizes performance.

Original architecture

Figure 9 shows the four layers of IBM's original architecture. The bottom layer includes a range of mobile and fixed devices; neither hardware architecture nor operating system must be homogeneous. The second layer contains device proxies, which every device has and which represent a transcoding layer for each device. The third layer is the user-proxy layer. Every user has a personal user proxy. This layer can store applications and a user's state. The fourth layer is the services layer, where the architecture implements shared applications, utilities, and servers. All requests between layers are in hypertext transfer protocol (HTTP). The requests can include data structures such as integers, characters, and strings. Each request includes user and device identification.

IBM implemented the architecture in Java. A device executes the service manager, which prepares requests and interprets responses for client applications running on the device. The device proxies use WEBI, an HTTP proxy that IBM developed. This proxy intercepts a user's requests, passes them through a series of user-specified filters, and forwards the transcoded requests and responses. The user proxy receives a request and either starts an appli-

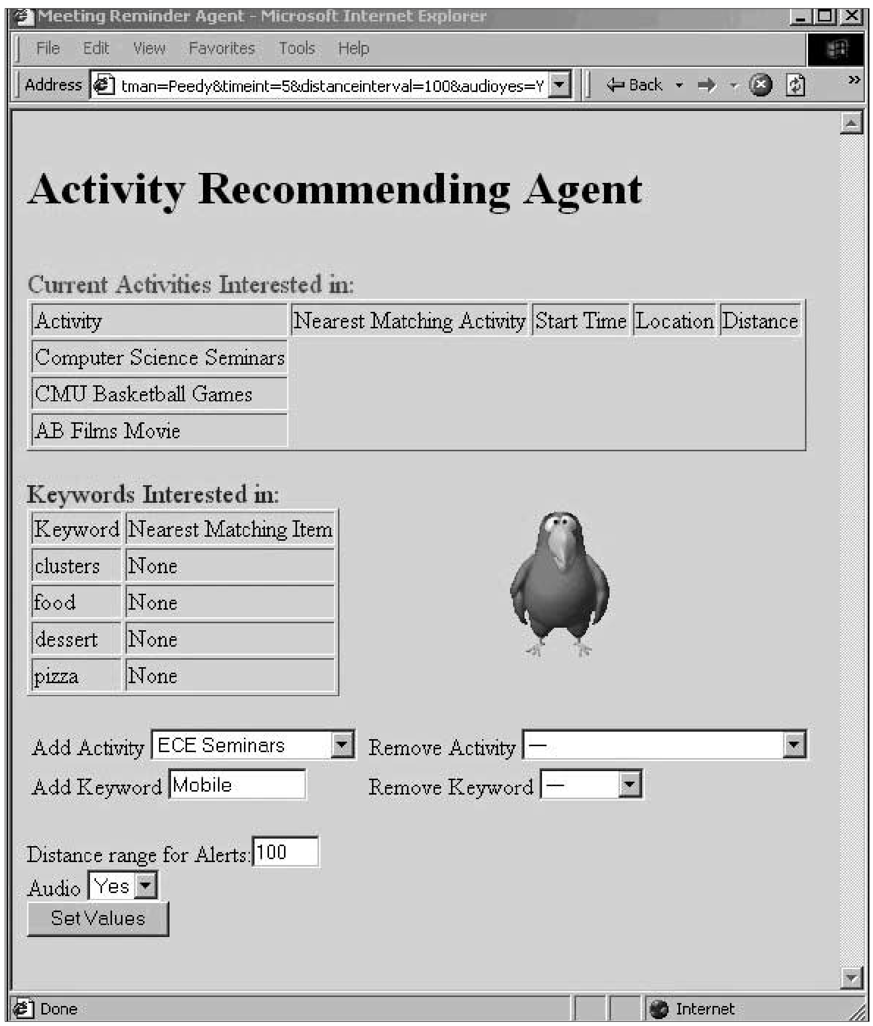


Figure 8. An example of an Activity-Recommendation Agent's user interface.

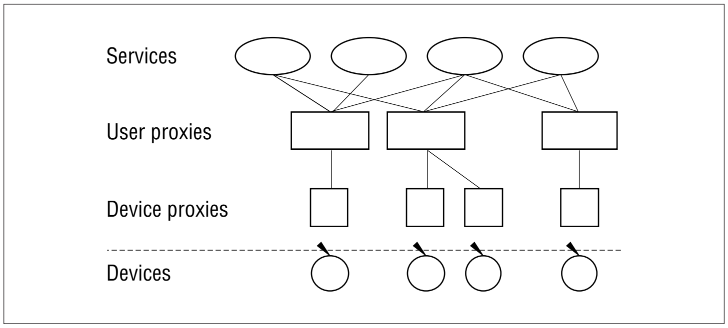


Figure 9. IBM's original architecture. Information passes from devices to services and back through device and user proxies.

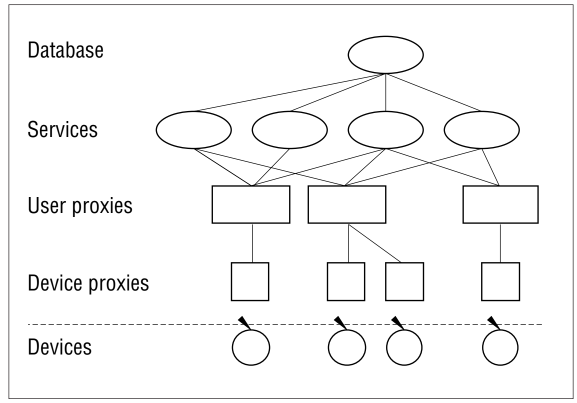


Figure 10. Handy Andy architecture. We added a fifth layer, a unified database that acts as a service, to IBM's original architecture (see Figure 9).

cation or forwards the request to a service. If the requested service is not well-known, or if the user's preferences don't define it, the user proxy invokes the Service Location Protocol to locate the requested service.

Revised architecture

The IBM architecture lets a service or user proxy store preferences and long-term state. We made the decision to create a unified (SQL) database as a service. We added a fifth layer to the architecture, above the services (see Figure 10). We named the new architecture Handy Andy, after Handheld Andrew, CMU's wireless-network project. In the Handy Andy architecture, all services' and user proxies' database access is based on the privileges of the user authenticated to them. That prevents common data such as a user's name, address, and contact information from being duplicated across systems. Users can update their data with a single application.

With the original architecture, conflicts pertaining to the format of stored information became apparent, as competing applications preferred certain data sizes and types. The revised architecture lets stored procedures translate stored data into any format that a service requests.

Idealink is a virtual meeting space tool. The user interface—a shared whiteboard that a user can archive for later review—is optimized for the minimal screen area that portable devices provide. The system operates within a client–server architecture: The application runs on the target devices, to which the server distributes screen updates. The Handy Andy architecture enables additional features and ease of implementation within a pervasive, wireless environment. Figure 11 shows a typical Idealink session's architecture elements within the Handy Andy architecture.

The Handy Andy architecture lets the system be more flexible. Also, the architecture automatically deals with problems inherent in wireless networks. Each user has one or more devices running the Idealink user interface. The devices might have color or black-and-white displays; their screen sizes might range from a watch-sized liquid crystal display to a wall-sized projected image. The architecture's device proxy adjusts color depth according to the device properties and instantiates filters that scale the size of screen updates. The devices do not use valuable clock cycles and battery power for these operations. If the communications channel between the device proxy and the device is broken, the device proxy caches updates until the device reestablishes connection. If the user so desires, he or she can start the Idealink session on one device and continue it on any other. From the user's calendar, the user proxy knows what meeting is taking place—which lets the system automatically negotiate who is included in the Idealink session. The user proxy stores preferences, including tool palette layout, and user-selected keystroke combinations. The Idealink service combines each user's additions to the session and updates each client. At the end of the meeting, the service archives the session in the database.

Location-sensing service

The Location Service generates a key parameter of context information. To determine a user's location, the wireless-network card (acting as a sensor) in the user's computer measures the signal strengths to all available wireless-access points and compares them to recorded training signal strengths. For every location, the sensor records a unique signal-strength reading from a group of access points. For training, the user manually inputs his or her location into the computer. The computer then takes and averages approximately 17 samples. This process generates a table that lists what signal levels to expect at different locations. The sensor requires only a single test, which it can save for use in later sessions and on other platforms. During use, the computer compares measured values to those in the table and computes differences. It reads the entry with the smallest difference as the current position.[2] As with Privacy Guard, the user requesting a target's location sends his or her request to a server. The server might use a caching mechanism to answer the request, or it might send the request to the target user. The target user's computer determines its location and sends the results to the server. The server completes the transaction by sending the target's location to the requesting user (assuming that user has permission to receive the target's information). Our Location Service is significantly more accurate than standard Global Positioning Systems.[2]

Table 1 presents the accuracy of our location-measurement results. We inferred accuracy from the fact that the distance needed for a signal-strength change of one decibel milliwatt (dBm) has been empirically determined to be approximately five feet when near an access point. Because more than 99.9 percent of our measurements are within 3 dBm of the actual value, we infer that the reported locations are within +/- 15 feet of actual positions.

Client–server speech issues

Speech-PHD requires significant computing resources for the automatic speech recognizer (CMU's Sphinx ASR—see www.speech.cs.cmu.edu for details) and for text-to-speech conversion (Festival Text-to-Speech software). When we were developing Speech-

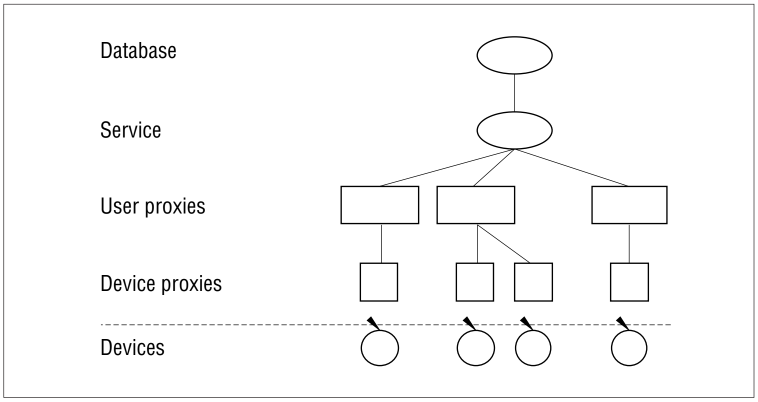


Figure 11. Idealink architecture within the Handy Andy architecture. The Handy Andy architecture provides a pervasive, wireless environment for the Idealink whiteboard tool.

PHD, no mobile device had the required computing power, memory, and nonvolatile storage. Placing the ASR and text-to-speech software on a server solved the resource limitations but introduced network latency. We measured latency from the end of the user's query until the system began its response. Table 2 summarizes the results. Transferring both the query and the response as a file required almost five seconds. By modifying Sphinx to stream the query, we reduced latency to two seconds. By modifying Festival to also stream the response, we reduced the delay by a factor of 25.

Lessons learned

The Handy Andy architecture provides a useful framework for developing persistent applications. The architecture is extremely broad in its description, letting developers implement very portable applications. A successfully implemented device proxy can maximize a device's usefulness while offloading expensive conversions to a network server. Developers can implement simple applications in the user proxy or as a service.

Developers have tried to make device proxies do more than they could. To fully exploit the capabilities of device proxies within the architecture, we explored speech- and user-interface-adaptation filters. ASR vocabularies are limited. Most require knowledge of the user's language in the form of a language model. The application must provide such a model; therefore, accessing the device proxies would no longer be transparent to the application programmer.

Our database lets users and services share data. Because we could customize the data-access interface for each service, we limited issues with proprietary application-protocol interfaces. The data inherited the database's security model, allowing user-permission specification and enforcement. This was convenient for programming, but it introduced a degree of failure: the database limited performance and exposed all shared data to security risks.

Table 1. Accuracy of location measurements.

Accuracy (%)	Strength (dBm)	Distance (feet)
68.6	+/– 0.939	+/– 5
95.4	+/– 1.146	+/– 10
99.9	+/– 2.817	+/– 15

Table 2. Speech-PHD network speech latency.

Latency (sec)	Transfer file query	Streaming query
Transfer file response	5	2
Streaming response	–	.2

Our proactive agents cannot access system-level functions, such as starting new applications, on the user's behalf. We won't let them do so until we address system-level security. Although context information helps generate more-intelligent system behavior, it is a liability for the system's users. Location information provides a prime example. All system levels, including the architecture, protocols, inferred preferences, and user-specified preferences, must address the security of such information.

We have not yet optimized the location service. Requiring the tracked client to return its current location for every request uses mobile devices' limited power and computation cycles. Ideas for increasing the location service's efficiency and scalability include caching and predicting user location. ■

Acknowledgments

We acknowledge the funding support of IBM Research, the US National Science Foundation under grant 9901321, and the Defense Advanced Research Projects Agency. We also acknowledge the students from the Rapid Prototyping of Computer Systems and Mobile Computing courses (Spring and Fall semesters) for their contributions to the project. We thank Mike Karasick for his continuous support to the project.

References

1. A. Smailagic and D. Siewiorek, "User-Centered Interdisciplinary Design of Wearable Computers," *ACM Mobile Computing and Comm. Rev.*, vol. 3, no. 3, July 1999, pp. 43–52.

2. J. Small, A. Smailagic, and D.P. Siewiorek, "Determining User Location for Context-Aware Computing through the Use of a Wireless LAN Infrastructure," submitted for publication to *ACM Mobile Networks and Applications*, vol. 6, 2001.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Authors

Joshua Anhalt is a graduate student of electrical and computer engineering at Carnegie Mellon University. His research interests include ubiquitous computing, context-aware computing, and the privacy of information. He received his BS in electrical and computer engineering from Carnegie Mellon University. Contact him at 5260 Forbes Ave., Pittsburgh, PA 15217-1102; anhalt@andrew.cmu.edu.

Asim Smailagic is a faculty member at Carnegie Mellon University's Institute for Complex Engineered Systems, College of Engineering, and Department of Electrical and Computer Engineering. He also directs CMU's lab for Interactive Computer Systems. His research interests include pervasive computing, system-level design of advanced computer systems, rapid prototyping of wearable computers, and audio-visual and sensor interfaces to computers. Contact him at Institute for Complex Engineered Systems, Carnegie Mellon Univ., Hamburg Hall, Pittsburgh, PA 15213; asim@cs.cmu.edu.

Daniel P. Siewiorek, Buhl University Professor of Computer Science and Electrical and Computer Engineering at Carnegie Mellon University, directs the Human–Computer Interaction Institute. His research interests are mobile computing, computer architecture, human–computer interactions, and reliability. He received his BS in electrical engineering from the University of Michigan and his MS and PhD in electrical engineering from Stanford. He is a member of the ACM, Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and the IEEE Computer Society. Contact him at Human–Computer Interaction Inst., CMU, Newell-Simon Hall, Pittsburgh, PA 15213; dps@cs.cmu.edu.

Francine Gemperle is a senior industrial designer with the Wearable Computing Group at Carnegie Mellon University. She is an expert in designing products for wearability and is working on becoming an expert in tactile interaction design. She received her BFA in industrial design from Carnegie Mellon University. She is a member of the Industrial Designers Society of America. Contact her at Carnegie Mellon University, Institute for Complex Engineered Systems, 2203 Hamburg Hall, Pittsburgh, PA 15213.

Daniel Salber is a research associate at the IBM T.J. Watson Research Center. He is interested in software engineering for human–computer interaction and context-aware computing. He received his MS and PhD in computer science from the University of Grenoble. He is a member of the IFIP Working Group 2.7 on Engineering for Human–Computer Interaction, ACM SIGCHI, and a founding member of AFIHM. Contact him at IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532; salber@acm.org.

Sam Weber is a research staff member at IBM's T.J. Watson Research Center. His research interests include software architecture, security, and verification. He received his BSc and MSc from the University of Toronto and MSc and PhD from Cornell University. His hobbies include swing and contra dancing and collecting mechanical puzzles. Contact him at IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532; samweber@watson.ibm.com.

Jim Beck is director of systems engineering and mobility at Inmedius Corp. and a visiting researcher at CMU's Human–Computer Interaction Inst. His research is in mobile computing, *n*-tier distributed applications, service frameworks, component-based software systems, embedded systems, computer architecture, design automation, hardware design, and digital application-specific integrated circuit design. He received his BS in electrical engineering from the Univ. of Pittsburgh, his MS in electrical engineering from Purdue, and his MS and PhD in electrical and computer engineering from CMU. Contact him at Inmedius, Inc., 417 S. Craig St., Pittsburgh, PA 15213; jbeck@inmedius.com.

Jim Jennings works on behalf of IBM in several international standards organizations. His research interests are in embedded computing. He received his PhD from the Computer Science Department of Cornell University. Contact him at IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532; jsj@acm.org.

Implementing Assistive Technology on Wearable Computers

David A. Ross, *Department of Veterans Affairs Medical Center*

What might seem perfectly intuitive to a young rehabilitation engineer designing assistive devices might not be intuitive at all to a disabled or elderly person experiencing a serious loss of function for the first time. When designers understand the complex nature of disabilities, they're more likely to meet the disabled users' needs.

The author presents his work in developing assistive technology for disabled users. He makes a case for wearable computers and focuses on design strategies that accommodate the users' changing needs.

Using the results of my work in designing assistive technology, this article describes impaired people's needs and offers design strategies to accommodate them. It also presents my research to develop a wearable computer-based orientation and wayfinding aid for the severely visually impaired.

The nature of disability

Approximately 54 million Americans are classified as disabled;[1] that is, they have a physical, cognitive, or sensory impairment—or a combination of the three—making it difficult or impossible to perform daily living activities. Disabilities might result from a disease, a birth defect, or a traumatic accident.

The disabled population is heterogeneous, with a wide range and variety of functional differences within each of the three types of functional impairments. For instance, sensory impairment might include loss of vision, hearing, smell, taste, touch, heat sensitivity, or pressure sensitivity. A single disease or accident can impact function in more than one of the three impairments. For instance, a spinal cord injury can cause the loss of both physical and sensate function.

The range and variety of functional losses increase with aging. Two-thirds of disabled Americans are over the age of 65.[1] As people with a primary disability grow older, many experience the onset of *comorbid* functional losses, such as the progressive loss of physical, sensory, or cognitive functions. Such comorbidities might drastically affect the aging person's ability to continue using his or her assistive devices. For instance, as a blind person with diabetes ages, he or she might experience progressive peripheral neuropathy, resulting in loss of touch in the fingers, making it difficult and eventually impossible to read Braille or use a tactile map.

The fastest growing disabled group is not people aging with a disability; rather, it is people aging into disability. These people have lived normal lives; however, as they pass the 60-year mark, they begin to experience age-related accidents and diseases.[1] Incurable age-related diseases exact a great toll. In the US, 40 percent of people over 65 are losing their vision as a result of glaucoma, macular degeneration, or diabetes.[2] Diabetes also causes atrophy of peripheral nerves in the hands and feet, resulting in a loss of sensation that can cause accidents and even loss of a hand or foot. Losing cognitive functions can lead to various stages of dementia and finally total loss of functional independence.

Older people who were once capable can find it difficult to learn to function independently with a disability. After retirement, people have less incentive to use unfamiliar technology or learn a new way of doing things. Also, third-party agencies are less willing to pay for a retired person's assistive technology, so the burden falls on the family. Furthermore, designers create most assistive technology for people with a single disability. Older persons with comorbid functional losses might thus find existing assistive technology difficult or impossible to use.

Because baby boomers, the largest segment of the US population, are now approaching their 60s, most Americans soon will be experiencing some degree of disability. So, personal technology that adapts to the

1094-7167/01/$10.00 © 2001 IEEE

Historical Perspective

Wearable assistive technology is centuries old and in most cases has progressed from handheld to orthotic to prosthetic. For instance, eyeglasses began as handheld devices, which gave way to lenses mounted into wearable frames, which gave way to contact lenses placed in the eye. Alexander Graham Bell invented the first viable electrical hearing aid—a handheld device. This eventually became a wearable device, and now it's implantable. Moving from handheld to wearable seems natural and desirable. People want to use assistive technology all the time and use it transparently while performing activities that occupy their hands and mind.

Historically, assistive technology was usually developed before personal technology. Bell developed the hearing aid before the telephone. The disabled's perceived needs always appeared greater than the consumer's seemingly more frivolous needs. In the early days, consumer telephones seemed frivolous and even undesirable.

In 1880, John Perry and W.E. Ayrton conceived of "electrical vision" using the same principle Bell used that year to invent the photophone wireless telephone. The conductivity of selenium was known to be light sensitive, and Perry and Ayrton took advantage of this property to develop a selenium receiver that blind people could wear on their foreheads. Headphones produced sounds proportional to the amount of light reaching the sensor. Blind people could become aware of objects in their surroundings that reflected and blocked light. The Elektroftalm (1897) and the Exploring Optophone (1912) were the first devices manufactured on the basis of this principle. In 1913, the Optophone was also used as a handheld reading aid.

The National Academy of Sciences, with support from the US Army and Veterans Administration, developed the first truly sophisticated wearable electronic aids in 1945 when they produced sensory aids for the blind. In the late 1950s and early 1960s, this research resulted in three products: the Pathsounder ultrasonic device, the C5 Laser Cane, and the Sonicguide. All of these were useful in dynamic travel situations.

In 1966, the Russell Pathsounder was produced, the first wearable assistive device that actually processed data before presenting it to the user. This device processed incoming sonic data to provide the user with a binary piece of information: the "path ahead is clear," or "not clear." The first use of an actual wearable computer as an assistive technology is unclear, because microprocessors have been used in assistive wearable technology since the 1970s. Based on functional electrical stimulation research performed at the Cleveland VA Medical Center, designers developed a wearable microprocessor-based device to control the stimulation of leg muscles in proper sequence to allow a paralyzed person to walk. Now that this technology is miniaturized, it can be implanted for long-term use.

I began using a wearable computer base (PC/104 hardware) in 1996 when I developed a remote-control device for Universal Voice Control of the Environment (U-Voice). U-Voice was designed for use by quadriplegics to give them the ability to verbally control devices around their homes using existing X-10 receivers available from Radio Shack and TV, VCR, and stereo systems' remotes. Essentially U-Voice was a universal remote control for people who can't push buttons. I integrated a PC/104 voice recognition board and a custom PC/104 board into this system to provide voice control of infrared remote and X-10-based devices. I used software that created an interface that was fully voice-interactive, except for the power switch. All interactions with U-Voice were through this interface, including learning how to use U-Voice and programming U-Voice to control a variety of devices.

aging user's changing needs will become very important over the next five to 10 years. The evolution of assistive devices is described in the "Historical Perspective" sidebar.

The changing needs of the disabled

Before developing assistive technology, designers must understand that disabled users' needs might change as they adapt to their condition and decide which activities they want to perform independently.

Adapting to a disabling condition takes time and involves a drastic transition between two ways of thinking, living, and functioning. What disabled people think they need to function can change dramatically as they reevaluate their priorities and values regarding what they want to be able to do. The adaptation time for this process varies. If their condition is stable, they might take a year or more to adapt. If their condition is progressive, as are most age-related debilitating diseases, they might never fully adapt.

Most experts agree that well-adapted people have successfully maximized their quality of life. They have accepted their limitations and pick and choose activities carefully, participating only in those about which they care most. They make choices by weighing the time and effort required for an activity against the perceived value of that activity, using a process similar to analyzing the return on an investment.[3] For maximizing quality of life, the general population might also find such a process useful.

For disabled people, the time and effort required to take care of basic necessities can be critical, because it directly affects the time they have for more valued activities. Recognizing this concern, many assistive-technology designers have focused on constructing devices to help people independently take care of their basic necessities. However, in a study of 200 visually impaired veterans in 1995, Atlanta Veterans Affairs investigators found that 60 percent of the veterans were using only approximately 40 percent of their assistive devices a year after they had received them.[4]

In a 1999 national survey on veteran satisfaction with VA rehabilitation services, the Atlanta VA investigators discovered the reason for such disuse.[4] They asked visually impaired veterans about their ability to independently perform such basic tasks as paying their bills and balancing their checkbooks. A statistically significant number responded that such tasks were very difficult or impossible. However, these veterans seemed to contradict themselves by stating that they were perfectly satisfied with their ability to function independently at home. Analysis revealed that in nearly all these cases a spouse or relative was performing these tasks for them.[4]

From this data, I would hypothesize that most disabled people choose caregiver assistance if they have a choice between doing necessary tasks with an assistive device or having a caregiver do such tasks for them. This is because the time and effort to independently perform such tasks is significant, even with a well-designed assistive device. This data confirms that well-adjusted people tend to make choices that minimize the time and effort they put into necessary, but mundane, daily activities so as to maximize time spent doing valued activities.

Assistive-device design goals can be moving targets when disabled people are in the adaptation phase. Likewise, designs for aging people with a progressive debilitating disease must have the flexibility to change with the user's needs. Understanding this, designers can choose a general-purpose base with interface hardware that can be adapted to those changing needs. Assistive devices must also be useful for performing a wide variety of functions because specialized single-use devices might go into disuse after a short time. Personal-technology designers have already become painfully aware of this problem in regard to the fickle nature of the consumer.

Assistive design strategies

Special-purpose devices designed to meet a small population's changing requirements would be expensive and perhaps discourage a manufacturer from producing them. Fortunately, by using wearable computers, designers can accommodate several assistive-technology designs in a single device and modularize the structure to make it amenable to a variety of needs and uses. The resulting device appeals to larger markets, costs less, and sells more easily.

Universal design, "an approach to creating environments and products that are usable by all people to the greatest extent possible,"[5] is a response to such issues. UD has seven principles:[5]

1. *Simple and intuitive use*. Is easy to understand, regardless of the user's experience, knowledge, language skills, or concentration level.
2. *Equitable use*. Does not disadvantage or stigmatize any group of users.
3. *Perceptible information*. Communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities.
4. *Tolerance for error*. Minimizes the consequences of accidental or unintended actions.
5. *Accommodation of preferences and abilities*. Accommodates a wide range of individual preferences and abilities.
6. *Low physical effort*. Can be used efficiently and comfortably and with a minimum of fatigue.
7. *Space for approach and use*. Provides appropriate space for approach, reach, and use regardless of the user's body size, posture, or mobility.

Adaptive design comprises a basic UD base platform and a variety of hardware modules and program modules that better meet the specific needs of a variety of users. It's such a new concept in assistive technology that only AD prototypes exist to date. AD extends UD principles and is applicable to cases where designers can't create a single device usable by everyone.

Clearly, the wearable computer is perfectly suited for this use. Unlike most closed handheld technology, wearable computers can accept a variety of accessible plug-in interfaces. When the user's needs change (for instance, a user with a slowly progressing disease), designers can add new applications to increase functionality. Wearable computers employing AD and UD principles would enable users to perform a wide variety of tasks hands-free with a single device. Furthermore, the wearable computer is always with the user, offering assistance whenever and wherever it is needed.

> By using wearable computers, designers can accommodate several assistive-technology designs in a single device and modularize the structure to make it amenable to a variety of needs.

Designing an orientation aid

In a one-year research project, my colleagues and I tested three different user interfaces on a *spatial orientation* and *wayfinding* aid that we have been developing. Our job was to determine which aspects of the interfaces were the most functional and intuitive.[6] We began the design process with a spatial orientation and wayfinding needs analysis of people aging with a severe vision loss. The people we tested were representative of the approximately 11.4 million visually impaired in the US, 10 percent of whom have no usable vision. Two-thirds of the visually impaired are 65 or older, and most experienced the onset of severe visual impairment after age 60.[2]

Spatial orientation is a major problem for people of all ages with severe visual impairment. It's particularly difficult for older people who might also be losing some cognitive, as well as proprioceptive (sensing the location, orientation, and movement of the body and its parts) and vestibular (maintaining equilibrium and balance) function. Spatial orientation is the ability to establish and maintain awareness of your position in space relative to landmarks in your environment and to your destination. It's distinctly different from mobility, which involves skillfully coordinating actions to avoid obstacles in your immediate path.[7]

Visually impaired people maintain spatial orientation by using cues detected by a number of senses; hearing is perhaps the most notable. When they walk near sound-making objects, they can perceive changes in spatial relationships with the shifting of sounds from those objects. They can determine the distance to a wall or doorway, for example, by listening to the echoes of sounds emitted by objects, including the sounds they're making themselves.

Acquiring the perceptual awareness and skills needed to maintain spatial orientation comes only with much practice, patience, and experience—even for young students with acute senses. Older adults with some hearing loss and perhaps other sensate losses might not be able to acquire all the skills needed to remain adequately oriented to the environment.

Wayfinding is how people use their spatial orientation to maintain a heading toward their destination regardless of the need to avoid obstacles. They need continuous feedback from the environment to guide their actions within perceived surroundings in dynamic settings (such as traffic intersections) because the tendency to veer from a straight path is a major problem.[7] Even if they are initially oriented to the environment, start out facing their destination, and encounter no obstacles, veering problems can necessitate frequent reorientation. A large body of research documents blind pedestrians' inability to maintain a straight-line path without external guidance. Even highly experienced blind pedestrians exhibit random errors large enough to make them veer into a parallel street on some occasions when crossing an intersection.[7]

Reviewing existing technologies

We critiqued existing technologies before establishing our design objectives and discovered much research in electronic travel aid development. Some viable handheld and wearable products were developed in the mid

’60s, and others have been developed more recently. The four different device types, with examples of available products, are

- *Single-output object preview*: Russell Pathsounder (wearable), Mowat Sensor (handheld), Polaron (handheld), Sensory 6 (wearable), and Walkmate (wearable);
- *Multiple output object preview*: Laser Cane (handheld cane);
- *Object preview with environmental information:* KASPA (previously the Sonicguide), vOICe, and BlindVision, (all wearable); and
- *Object preview with artificial intelligence*: Sonic Pathfinder (wearable).[8]

These types of devices, though, do not provide orientation and wayfinding information. We found relatively little comparable research in general-purpose orientation aids and discovered that the few existing orientation devices have many limitations. None offers a full complement of the orientation information most often needed, which includes

- current location and heading relative to known landmarks and the desired destination;
- distance and direction to surrounding landmarks and the desired destination;
- overall layout of the greater surrounding environment; and
- things of particular interest to the user in both the proximate and greater surrounding environment.

The earliest approach for describing the immediate environment’s layout was Braille labels. However, visually impaired people have no means of knowing whether or where these labels are in a particular environment. Designers have developed technology for producing Braille and tactile maps. However, they’re bulky and not easily carried for in situ reference, and older people might have difficulty using them.

The Smith-Kettlewell Eye Research Institute developed Talking Signs to use in addition to Braille labels. These signs employ a coded light beam to transmit a set message from the sign to a handheld receiver, providing information typically found on printed signs. Furthermore, people can use the light beam as a beacon to orient themselves. Special versions of these signs can be integrated into pedestrian crossing signals, providing “Walk,” “Don’t Walk,” and “Don’t Start” information, as well as orienting users to the opposite corner’s direction.

Talking Sign systems have a major limitation. To discover the sign’s existence, the user must regularly scan new settings with the handheld receiver. However, for safety reasons, the manufacturers recommend the user not walk while using the receiver, which limits its use to familiar settings. The system does nothing to prevent the user from veering while walking. Also, handheld receivers make it difficult to carry anything because one hand is already occupied with a cane.

Verbal Landmarks is a wayfinding tool that uses inductive-loop radio signals. The user hears a spoken message when the belt-worn or pocketed receiver is within five feet of the transmitter. While the Talking Sign system relies on line-of-sight transmission, the inductive loop transmits through obstacles and often broadcasts extensive messages offering a more comprehensive introduction to the environment.

> For a visually impaired person, walking a straight line across the street is very difficult, especially if there is no parallel traffic or if the person is hearing impaired.

Mike May of the Sendero Group developed Atlas Speaks and GPS Talk (formerly called Strider). Atlas Speaks is a talking map for PCs that people can use to orient to a location before venturing out. GPS Talk is a more general-purpose device that employs a laptop computer in a backpack to integrate Atlas Speaks, a Global Positioning System (GPS) receiver, and a digital compass into a single portable device that can provide in situ information about the user’s location and heading, a particular destination’s direction, and information about the surrounding environment.

However, although GPS Talk represents a major step toward a general-purpose orientation device, there is much it does not do. Because it relies on GPS, it does not function indoors, and outdoors it provides insufficient resolution for users to easily locate a doorway into a building. Even with the declassification of the GPS signal, GPS Talk claims only 10-meter accuracy. This is insufficient to direct users across the street with the assurance that they will be able find the opposite corner. Furthermore, in large cities where buildings might block the line of site to four GPS satellites, obtaining position information is difficult.

In addition, GPS Talk does not interact with devices in the environment to provide temporal information such as the state of a traffic light. Finally, GPS Talk’s speech interface might not be usable by people with hearing loss and might not be appropriate in noisy outdoor settings.

The other missing element in these products is a highly functional and intuitive user interface. How do designers create an interface that is easy to use, moves unobtrusively with the user, and provides the type of feedback needed in a variety of settings?

This is a question Jack Loomis, a psychology professor at the University of California, Santa Barbara, has been attempting to answer the last 14 years. To assist blind persons, Loomis is developing a virtual sound interface in which the objects and buildings around the user verbally identify themselves seemingly from their actual location in 3D space. This interface uses GPS, Geographic Information System data, and stereo headphones. He has learned that simulating 3D spatial location using sound is difficult when using headphones. However, he hopes to solve this problem by developing more sophisticated sound-processing algorithms. You can access his papers at www.psych.ucsb.edu/~loomis.

Evaluating user-friendly interfaces

In 1997, I evaluated three orientation device designs for indoor use in a project, “Cyber Crumbs: Subject Testing an Orientation Aid for Veterans with a Visual Disability.”[6] My colleagues and I tested 20 severely visually impaired older adults. Results showed that tiny digital transmitters at hallway intersections reliably provided needed orientation information. However, most subjects commented on the user interface rather than the location technology. So, our next study focused on the interface instead of the orientation technology.

Also, instead of testing the products indoors, we evaluated them in a more demanding situation—street crossing. To

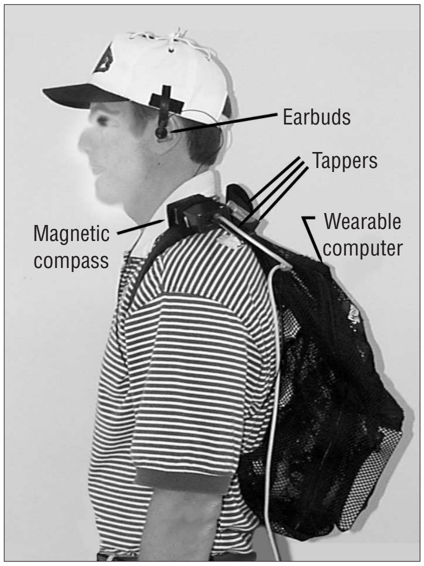


Figure 1. A prototype assistive device for guiding the visually impaired.

Figure 2. Subject crossing street wearing the orientation device.

cross a street, a visually impaired person performs four critical tasks:

- detecting the street or curb,
- aligning the body with the edge of the curb facing the opposite corner,
- initiating crossing at the proper time, and
- walking a straight path across the street to the opposite corner.[7]

All these tasks have become more problematic in recent years. Walking a straight line across the street is very difficult, especially if there is no parallel traffic or if the person is hearing impaired.

Because street crossing is one of the most difficult, hazardous, critical, and crucial tasks we might select for testing an orientation aid, it provided the opportunity to fully test interface viability. The rationale was that if our subjects felt confident and safe using our orientation aid during street crossing, the device would most likely suit their needs in less crucial settings as well.

Developing a prototype

We designed and constructed a prototype device that integrated three interfaces:

- a virtual 3D sonic beacon marking the opposite street corner,
- a spoken heading directing users to the opposite street corner, and
- a tactile shoulder tapper indicating the direction of the opposite street corner.

For expedience, we provided a digital compass to determine the user's orientation in the test site intersections and provide feedback. However, I recommend that a compass not be the only basis of an orientation aid. We had to calibrate the magnetic compass for use at each intersection and recalibrate it at the start of each crossing to maintain the accuracy needed to guide the person reliably to the opposite corner. In addition, stuff below the streets can deflect (or attract) the magnetic compass, making it very unreliable.

We built the wearable computer base with boards manufactured by Adaptive Systems. It contained a 66-MHz 486 CPU with 16 Mbytes of RAM, a 200-Mbyte hard drive, an I/O card with two serial ports, and a SoundBlaster card for stereo sound presentations. We used Windows 95 to exploit its 3D sound generation modules. We provided software to

- interpret digital compass data arriving through the serial port,
- drive the three different interfaces, and
- implement and automate subject-testing procedures.

The virtual beacon. We used Windows sound modules and the SoundBlaster card to perform virtual-beacon presentation. The card produced a recorded bell-like tone. We located this "bell" relative to the user's heading by employing data from the digital compass (perched on the shoulder—see Figure 1) and known data about the widths of the test site intersections. We updated the bell location values approximately 30 times per second, so that perceptual latency was minimal. Although the bell sounded only once every two seconds, the decaying tone tracked subject movements. A pair of earbuds (earphone pieces that fit into the ear but not inside the ear canal) was located on the cap and provided the stereo output (see Figure 1). Users could adjust the earbuds to rest approximately one-half inch in front of the ear canal so that they could still easily and naturally hear subtle environmental sounds.

Spoken headings. This interface generated digitized speech through the SoundBlaster card to the earbuds. We developed software to convert digital compass data into clock face positions. The user could hear the relative position of the destination (for example, "one o'clock") announced once every two seconds.

The tapper. This device employed three small contact speakers that lightly tapped the user. When the user was heading in the correct direction, a speaker at the back of the neck activated. When the user needed to turn right or left to line up with the target, a speaker on the right or left shoulder activated. The tapper activated twice a second; we made its repeat time faster than the speech generator's to compensate for its limited spatial resolution.

Subject testing protocols

We recruited and tested 15 subjects. We recorded their visual pathology and any age-related comorbid pathologies. We tested them at three intersections near the Atlanta VA Medical Center. During these tests, we let the subjects use their canes but not dog guides. Figure 2 shows a subject crossing the street using the orientation device.

After a baseline pretest, we fitted our subjects with the device and trained them on each interface in random order. The subjects then used each interface in random order to cross each intersection both forward and back. Following the device tests, the subjects removed the prototype and crossed each intersection both forward and back in a baseline post-test. We measured crossing time, off-target error, out-of-crosswalk errors, hesitations, and any apparent subject confusion.

After the tests, we asked the subjects to rank the interfaces from the most to the least useful. Then we asked them whether any of the interfaces helped them find their way across the street better than using their canes or dogs. If they reported that an interface performed better, we asked them how it was better. If they reported that it didn't, we asked them what made it difficult to use. Finally, we asked them for ideas on how to improve each interface, and, given this improvement, what interface or interface combinations they would then prefer.

Subject demographics

Our subjects ranged in age from 62 to 80; the average was 68. Their condition ranged from totally blind for over 40 years to partially sighted, with the best acuity being 20/300. Over one-half of the subjects were totally blind. Our subjects' frequency in crossing streets ranged from a few crossings a week to several a day. They crossed streets ranging from low-volume streets close to home to high-traffic streets some distance from home. Their independence level ranged from almost always crossing with someone else to almost always crossing by themselves. Two subjects had dog guides; the others used canes.

Data analysis

We converted street-crossing times to walking pace in feet per second. We converted target errors to inches of veer per foot forward. We calculated average "normal" pace and veer for each subject from pre- and post-baseline measures. We calculated the ratio of prototype performance (pace and veer) to baseline performance for each subject with each interface. We used these ratios as relative indicators of each subject's performance improvement. We performed standard t-tests to determine the significance of performance improvements for each interface. We used subject rankings to produce weighted "votes" for each interface. We used t-tests to identify significant differences in the vote tallies. Finally, we grouped subject critiques and comments by type—comment, criticism, or improvement idea—and tallied them.

Results

Performance varied widely, with some significant differences for the different interfaces. Most subjects achieved best performance using a particular interface; however, the actual interface that resulted in the best performance varied from person to person. On the basis of the mode of operation that most improved each subject's performance, we compared performance with and without the prototype.

> Most subjects achieved best performance using a particular interface; however, the actual interface that resulted in the best performance varied from person to person.

Table 1 shows no significant improvement in walking pace when the subjects used their best interface; however, their veering performance improved significantly, with the average veer reduced to 31 percent of the baseline veer. We find this not only statistically significant but also quite meaningful. The average baseline veer was 10 feet, which is enough to cause the person to miss the opposite curb and walk into the center of the intersection. On the other hand, average veer with the best interface was only 3 feet, which brought the person close enough to the curb to easily find it with a cane.

Furthermore, when the subjects used their best interface, their number of hesitations, amount of confusion, and movement out of the crosswalk, compared with baseline measures, decreased to one-third. We also noted that when particular subjects did not like an interface, their performance decreased relative to the baseline measures.

Table 1. Performance for subjects' best mode of operation.

Measure	Change	Significance
Walking pace	1.04	None
Veering	0.31	.001

From both the objective measures and the subjects' responses, we discovered the tapping interface worked best for one-half of the subjects and the 3D virtual beacon worked best for the others. All the subjects disliked the speech interface because of the speech feedback's timing. They said the speech interface was simple and easy to respond to, but the two-second interval was too long to tell them if they were veering. Also, the subjects tended to overcorrect when told they were off course.

Subjects who preferred the virtual beacon commented that, unlike the other interfaces, it did not cause them to overcorrect and that they didn't have to concentrate a lot to use it. Subjects who preferred the tapping interface commented that it didn't make them stand out like the headset did or interfere with hearing traffic sounds. They liked that they could feel it even when they couldn't hear anything because of the loud traffic.

When asked which interface they would prefer if their suggested improvements were made, six subjects chose the speech interface, five chose the tapping interface, and four chose the virtual sound beacon. Four subjects also suggested that a combination of speech and tapping interfaces would be ideal; two volunteered that a combination of the speech and virtual beacon interfaces would be ideal.

The improvement they most often requested was to make the speech interface more immediately responsive. They wanted it to tell them the instant they got off course, and then to verify that they were back on course the instant they were; and otherwise to shut up. They said they'd like to have an automatic sound level adjustment with ambient sound levels for both speech and the bell tone, and the ability to switch among these interfaces to get the one they wanted when they wanted it. They also wanted us to modify the tappers to be more insistent when they got off course, to tap immediately (in less than a half second) in the center when back on course, and then to stop tapping if they stayed on course.

Project conclusions

Given the subjects' comments, we were unable to establish a clear interface "winner." Perhaps we should not have asked, Which interface is the best? Maybe we should have instead asked, How can we optimize and modularize the interfaces so that users can easily assemble an overall interface that best suits their needs and preferences?

So, we concluded that each interface has a clear role in helping people with severe visual disabilities walk a much straighter path across the street, which also means walking a straight path along planned routes. The most statistically significant result is the decrease in veering, which will allow a person using a cane to find particular locations, such as doorways, and stairs, in addition to curbs.

We also concluded that optimizing the timing of speech feedback could considerably improve the speech interface. Given such improvement, speech output could be as viable as the other interfaces. On the basis of this, we recommend optimizing the speech interface further, and designing and testing a modular system that offers user-selectable combinations of optimized versions of these three interfaces.

Designers can apply the strategies for creating assistive technology directly to the design of technology for people who must work under disabling conditions (for example, firefighters who can't see because of thick smoke, construction workers who can't hear or verbally communicate when using noisy equipment, or astronauts who have their physical movements severely restricted by a spacesuit). This rationale also can be extended to the design of personal technologies for people on the go who want unobtrusive functional assistance in situ at a moment's notice. Inventive wearable computer designs, interfaces, and applications for the disabled will lead to applications and interfaces that the general population will want, and wearable computers will become the preferred personal technology devices.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Author

David A. Ross has been a rehabilitation engineer for 18 years. He began work in this field as a master's student at the Georgia Institute of Technology under the auspices of the Office of Interdisciplinary Programs, which was developing a funded program in rehabilitation engineering. In 1983, he became a full-time investigator at the Atlanta VA Rehab R&D Center, where he became a prominent principal investigator of federally funded rehabilitation technology projects with a primary focus in visual disabilities and aging. Over the past six years, he has developed wearable assistive technology employing wearable computers, and he plans to expand his work in this area considerably with the development of a universally interactive wearable aid. He holds a MEd from Georgia State University and an MS in electrical engineering from the Georgia Institute of Technology. Contact him at the Atlanta VA Rehab R&D Center, Dept. of Veterans Affairs Medical Center, Atlanta, GA 30033; davidross1@mindspring.com.

Acknowledgments

The Department of Veterans Affairs Rehabilitation Research and Development Service funded this research. The Atlanta VA Rehabilitation R&D Center is part of the Atlanta VA Medical Center Research Service, which provided laboratory space and equipment, administrative support, and staff resources for this research.

References

1. J.M. McNeil, "Americans with Disabilities: 1994–95," *US Bureau of the Census: Current Population Reports*, US Government Printing Office, Washington D.C., 1997.
2. *Estimated Statistics on Blindness and Vision Problems*, Nat'l Soc. to Prevent Blindness, New York, 1979.
3. G.W. Kelly, *The Participants' Description of the Personal Adjustment Process to the Disability Experience*, UMI, Ann Arbor, Mich., 1992.
4. W.R. De l'Aune, M.D. Williams, and R.L. Welsh, "Outcome Assessment of the Rehabilitation of the Visually Impaired," *J. Rehabilitation Research and Development*, vol. 36, no. 4, Oct. 1999.
5. B. Connell et al., "Development and Validation of Principles of Universal Design," *Proc. RESNA '96 Ann. Conf.*, RESNA Press, Washington, D.C., 1996, pp. 450–452; www.dinf.org/resna96/page167.htm (current 11 June 2001).
6. D.A Ross and B.B. Blasch, "Wearable Interfaces for Orientation and Wayfinding," *Proc. 4th ACM SIGCAPH Conf. Assistive Technologies* (ASSETS 2000), ACM Press, New York, 2000, pp. 193–200.
7. D.A. Guth and J.J. Rieser, "Perception and the Control of Locomotion by Blind and Visually Impaired Pedestrians," *Foundations of Orientation and Mobility*, 2nd ed., AFB Press, New York, 1997, pp. 9–38.
8. L.W. Farmer and D.L. Smith, *Adaptive Technology: Foundations of Orientation and Mobility*, 2nd ed., AFB Press, New York, 1997, pp. 231–259.