Chapter 20

Wearing It Out: First Steps Toward Mobile Augmented Reality Systems

Steven Feiner
Blair MacIntyre
Tobias Höllerer

Columbia University, U.S.A.

20.1 Introduction

Over the past decade, there has been a ground swell of activity in two fields of user interface research: augmented reality and wearable computing. *Augmented reality* [1] refers to the creation of virtual environments that supplement, rather than replace, the real world with additional information. This is accomplished through the use of "see-through" displays that enrich the user's view of the world by overlaying visual, auditory, and even haptic, material on what she experiences. Visual augmented reality systems typically, but not exclusively, employ head-tracked, head-worn displays. These either use half-silvered mirror beam splitters to reflect small computer displays, optically combining them with a view of the real world, or use opaque displays fed by electronics that merge imagery captured by head-worn cameras with synthesized graphics. *Wearable computing* moves computers off the desktop and onto the user's body, made possible through the miniaturization of computers, peripherals, and networking technology. (While we prefer this general definition implied by the

term "wearable," some researchers have favored a more restrictive definition, requiring, for example, a capability for hands-free operation [2] or a physical appearance that makes the user and others consider the computer to be part of the user [3].)

We are interested in how augmented reality can be combined with wearable computing, with the ultimate goal of supporting ordinary users in their interactions with the world. To experiment with these ideas, we have been building the software infrastructure and application prototypes described in this chapter. We believe that the commercial success of future descendants of these systems will depend in large part on hardware issues, including the quality, size, comfort, and cost of wearable displays and computers. However, our research focus has been on developing experimental software infrastructure and user interfaces. To maintain this emphasis, we use commercially available hardware exclusively. Furthermore, rather than using a commercial belt-size wearable computer, in one project we rely on a much larger back-pack–sized computer. While this decision sacrifices current user comfort, in return we get access to the same processor, memory, operating system, and busses that we would expect on a desk-top system instead of the compromises imposed by the smaller form factor.

The applications that we address here provide users with information about their surroundings. One system presents a construction worker with instructions for assembling a modular building, while another creates a personal "touring machine" that assists the user in exploring our campus. There are two themes that we have stressed in this research:

- Presenting information about a real environment that is integrated into the 3D space of that environment.
- Combining multiple display and interaction technologies to take advantage of their complementary capabilities.

In Section 20.2 we discuss related work. Sections 20.3 and 20.4 describe our construction assistance and touring machine prototypes, respectively, including pictures generated by our testbed implementations. In Section 20.5, we describe our system-design approach and the hardware and software used. Finally, Section 20.6 presents our conclusions and future directions.

20.2 Related Work

Previous research in augmented reality has addressed a variety of application areas including aircraft cockpit control [4], assistance in surgery [5], viewing hidden building infrastructure [6], maintenance and repair [7], and parts assembly [8]. In contrast to these systems, which use see-through head-worn displays, Rekimoto [9] has used hand-held displays to overlay information on color-coded objects. Much effort has also been directed towards developing techniques for precise tracking using tethered trackers (e.g., [10]–[13]).

Work in wearable user interfaces has included several projects that allow users to explore large outdoor spaces. Loomis and his colleagues have developed an application that makes it possible for blind users to navigate a university campus by

tracking their position with differential GPS and orientation with a magnetometer to present spatialized sonic location cues [14]. Petrie et al. have field-tested a GPS-based navigation aid for blind users that uses a speech synthesizer to describe city routes [15]. The CMU Wearable Computer Project has developed several generations of mobile user interfaces using a single hand-held or untracked head-worn display with GPS, including a campus tour [16]. Long et al. have used infrared tracking in conjunction with hand-held displays [17]. Mann [3] has developed a family of wearable systems with head-worn displays, the most recent of which uses optical flow to overlay textual information on automatically recognized objects. Starner [2] has explored the potential uses of wearable computers that can recognize what a user is doing to provide useful information to collaborating team mates.

Prior to the development of VRML, several researchers experimented with integrating hypertext and virtual environments [18]–[20]. All investigated the advantages of presenting hypertext on the same 3D display as all other material, be it head-worn or desktop. In contrast, some of the work described here exploits the different capabilities of our displays by presenting hypertext documents on a relatively high-resolution 2D hand-held display, which is itself embedded within the 3D space viewed through a lower-resolution head-worn display [21].

20.3 ARC: Augmented Reality for Construction

ARC (Augmented Reality for Construction) addresses the construction of spaceframe buildings [22]. Spaceframes are made from a large number of modular components, typically cylindrical struts and spherical nodes. While the external appearance of many pieces may be identical for aesthetic reasons, the forces they will bear, and their inner diameters, vary depending on their position in the assembled structure. Thus, it is possible to assemble pieces in the wrong position, which, if undetected, could lead to structural failure. Furthermore, if too many or too few pieces are put together in a subassembly that is to be hoisted into its final position, it may break apart in mid-air. While workers who are trained on a particular spaceframe system are familiar with how to assemble individual components, the position and order in which components are assembled differ from one structure to another. ARC is an attempt to prototype a system that could guide construction workers to put the right piece in the right place at the right time. While the tracking system that we used (see Section 20.5) required a tethered implementation, and our demonstration venues were indoors, we designed our user interface for a mobile outdoor environment.

We have been working with a diamond-shaped, full-scale aluminum system, shown in Figure 20.1, manufactured by Starnet International. We taped to each physical component a set of labels that identify the component using both a human-readable number and a machine-readable barcode. The user wears a see-through head-worn display with integral earphones and a tool belt that holds a barcode reader. The spaceframe is assembled one component (strut or node) at a time. For each component, ARC performs a series of steps.

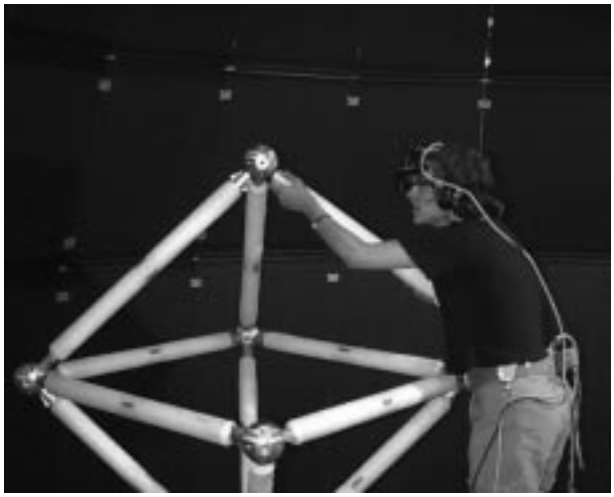


Figure 20.1 Prototype construction assistance system. The user wears a see-through head-worn display and a tool belt that serves as a holster for a hand-held barcode reader. Here, the user is following instructions to install a strut.

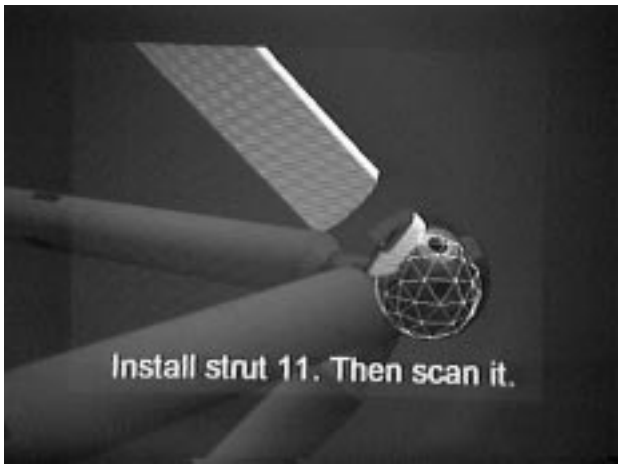


Figure 20.2 Prototype construction assistance system. User's view through the see-through head-worn display for the task shown in Figure 20.1. The user sees two real struts and a real node overlaid with a virtual strut (top) and textual instructions that instruct the user to install the strut in the location shown. A rotating arrow indicates the direction in which a fastener must be turned, and a wireframe sphere highlights the node to which the strut will be attached.

First, the worker is directed to a pile of parts and told which one to pick up. This is done by displaying textual instructions and playing a sound file of verbal instructions. Next, ARC confirms that the user has the right piece by having her scan a barcode on it. ARC then directs her to install the component, as shown in Figure 20.1. Overlaid graphics, such as those of Figure 20.2, indicate the correct location, and verbal instructions explain how to install it.[1] Finally, ARC verifies that the right piece is in the right place by asking the user to scan the component with the tracked barcode scanner (Figure 20.3).

Figure 20.3 Prototype construction assistance system. The user scans one of the barcodes on a strut with a position-tracked barcode reader, to verify that the right strut has been installed in the right place.

ARC was first demonstrated at the *ASCE Third Congress on Computing in Civil Engineering* in June 1996, where dozens of attendees tried the system. Based on this experience, we modified the user interface in two ways for a demonstration given at the *ACM '97* conference in March 1997.

First, while construction workers are familiar with the convention of attaching a fastener nut by turning it clockwise, many of our demonstration participants were not. Some had difficulty following verbal directions describing the direction in which to turn a fastener or did not understand that a strut was to be fastened to a node. Therefore, we added to the graphics a rotating arrow and a representation of the node(s) to which a strut was to be attached, shown in Figure 20.2, effectively solving the problem.

Second, other attendees could see only the participant and, on a separate monitor, the participant's overlaid graphics. Since we used an optical see-through display, many attendees found the graphics uninteresting in isolation and had difficulty understanding that the participant was seeing the graphics overlaid on the real world.

[1] Figures 20.2, 20.5, and 20.6 were imaged with a dummy head whose right eyesocket contains a Toshiba IK-M41A 410,000 pixel miniature color CCD camera used to record through an eye of the see-through head-worn display.

We rejected the approach of displaying to the other attendees a synthesized composite of the real and virtual objects in the participant's view, on the grounds that it would be confusing—we were trying to emphasize that the participant's view was being augmented, not replaced. Instead, we created an additional, stand-alone display that provided a third-person remote supervisor's view of the task. Here, the participant is shown as a tracked mannequin, surrounded by the structure under construction and overlaid with information about the current task being performed. Attendees viewed this overview on a large projection display and could also see the participant's graphics on a separate display.

20.4 A Touring Machine

Our "touring machine" prototype assists a user exploring Columbia's campus, overlaying information about items of interest [23]. As a user moves about, she is tracked through a combination of satellite-based, differential GPS (Global Positioning System) position tracking and magnetometer/inclinometer orientation tracking. Information is presented and manipulated on a combination of a head-tracked, see-through, head-worn, 3D display, and an untracked, opaque, hand-held, 2D display with stylus and trackpad.

Consider the following scenario, whose figures were created using our system. The user is standing in the middle of our campus, as shown in Figure 20.4. His tracked see-through head-worn display is driven by a backpack computer, and he is holding a hand-held computer and stylus.

Figure 20.4 Prototype campus information system. The user wears a backpack and see-through head-worn display, and holds a hand-held display and stylus.

As the user looks around the campus, his see-through head-worn display overlays textual labels on campus buildings, as shown in Figures 20.5 and 20.6. (These images were shot through the head-worn display, and are somewhat difficult to read because of the low brightness of the display and limitations of the video recording technology.) Because we label buildings, and not specific building features, the relative inaccuracy of the trackers we are using is not a significant problem for this application.

Figure 20.5 View shot through the see-through head-worn display, showing campus buildings with overlaid names. Labels increase in brightness as they near the center of the display.

Figure 20.6 A view of the Philosophy Building with the "Departments" fly-down menu item highlighted. Since the "Departments" menu item has been selected, a copy was animated flying to the bottom of the display to indicate the presence of new information on the hand-held computer, and the department list for the Philosophy Building was displayed around the building's name.

At the top of the display is a menu of choices: "Columbia:", "Where am I?", "Depts?", "Buildings?", and "Blank". When selected, each of the first four choices sends a URL to a web browser running on the hand-held computer. The browser

then presents information about the campus, the user's current location, a list of departments, and a list of buildings, respectively. The URL points to a custom HTTP server on the hand-held computer that generates a page on the fly containing the relevant information. The generated pages include links back to the server itself and to pages anywhere on the world wide web, to which we are connected through radio modem. The last menu item, "Blank", allows the head-worn display to be blanked when the user wants to view the unaugmented campus. Menu entries are selected using a trackpad mounted on the back of the hand-held computer. The trackpad's x coordinates are inverted to preserve intuitive control of the menus.

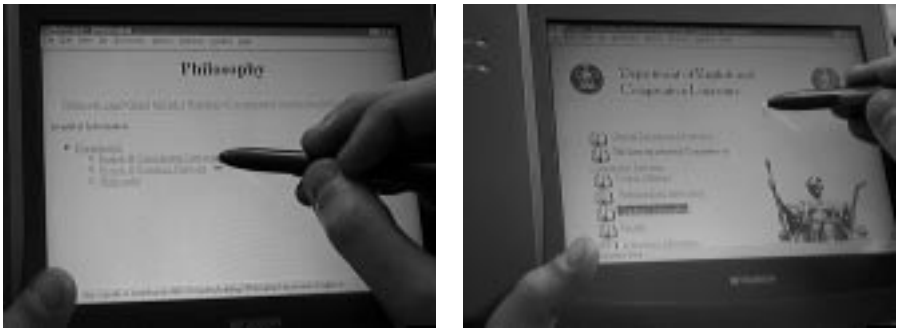
Labels seen through the head-worn display are grey, increasing in intensity as they approach the center of the display. The one label closest to the center is highlighted yellow. If it remains highlighted for more than a second, it changes to green, indicating that it has been selected, and a second menu bar is added below the first, containing the name of the selected building and entries for obtaining information about it. A selected building remains selected until the user dwells on another for more than a second as indicated by the color change. This approximation of gaze-directed selection can be disabled or enabled via a trackpad button.

When a building is selected, a conical green compass pointer appears at the bottom of the head-worn display, oriented in the building's direction. The pointer turns red if the building is more than 90 degrees away from the user's head orientation (i.e., behind the user). The pointer is especially useful for finding buildings selected from the hand-held computer. To do this, the user turns off gaze-directed selection, displays a list of all buildings via the "Buildings?" top-level menu entry, selects with a stylus the building he is looking for on the hand-held computer, and then follows the direction of the arrow pointer to locate that building. When the building's link is selected on the hand-held computer, the system immediately reflects the selection on the head-worn display. This is made possible by our custom HTTP server, which can interact with the backpack computer on URL selection.

A building's menu bar contains the name of the building plus additional items: "Architecture", "Departments", and "Miscellaneous". Selecting the building's name from the menu using the trackpad sends a relevant URL to the hand-held computer's browser. Selecting any of the remaining menu entries also sends a URL to the browser and creates a collection of items that are positioned near the building on the head-worn display. These items represent the information requested by the second-level menu entry selection and they stay in the same position relative to the building (and its label) until this menu level is left via a "Dismiss" entry.

When menu items that send URLs are selected, a copy of the menu item is translated down to and off the bottom of the head-worn display. This animated "fly-down" menu is intended to call the user's attention to the new material on the hand-held computer. For example, Figure 20.6 shows the Philosophy Building with the "Departments" menu item highlighted after its selection and animation has been completed. The building is annotated with the names of its departments and the hand-held computer's automatically-generated web page is shown in Figure 20.7(a).

Information about the selected building can be accessed in two ways. On the head-worn display, the user can select one of the surrounding items with the trackpad to present relevant information about it on the hand-held display. Alternatively, the user can select a corresponding item from the automatically-generated web page.



(a) (b)

Figure 20.7 (a) Selecting the "Departments" menu item causes an automatically-generated URL to be sent to the web browser on the hand-held computer, containing the department list for the Philosophy Building. (b) Actual home page for the English and Comparative Literature department, as selected from either the generated browser page or the department list of Figure 20.6.

For example, Figure 20.7(a) shows the URL selection for the English Department in the Philosophy Building, resulting in the display of its regular web page in Figure 20.7(b).

Another way of accessing information about a specific department is through the global list of departments that is produced on the hand-held by selecting the top-level "Departments?" menu item on the head-worn display. In this case the associated building does not have to be selected beforehand.

20.5 System Design

The following subsections describe some of the hardware and software decisions we made in designing our prototypes.

20.5.1 Hardware

Computers. Given the relatively small amount of overlaid graphics displayed in the ARC project, we were able to support the worker's view with a 90MHz Pentium PC, rendering entirely in software using Criterion RenderWare. In contrast, the supervisor's view is completely synthesized and includes a relatively complex view of a larger structure of which our demonstration's components are a subset; it required higher-end SGI and Sun workstations to achieve respectable performance.

Our touring machine's original backpack computer, obtained in early 1996, was a Fieldworks 7600 with a 133MHz Pentium processor and a cage that can hold 3 ISA and 3 PCI cards. While it was a big compromise in weight and size, it has significantly simplified our development effort, for example allowing us to use 3D graphics cards, such as our initial selection, the Omnicomp 3Demon OpenGL card, based on the Glint 500DTX chipset. We are currently replacing this unit with a

smaller, more powerful, laptop. The hand-held computer is a Mitsubishi Amity SP, which has a 75MHz DX4, and 640×480 color display, and integral stylus. Control of the head-worn display menu is accomplished through a Cirque GlidePoint trackpad that we mounted on the back of the Amity.

Head-worn display. Both applications use the Virtual I/O i-glasses optical see-through head-worn display, a relatively inexpensive, but relatively low resolution (60,000 triads), color display. We have also experimented with a Virtual I/O 640×480 resolution greyscale display.

Trackers. Given the large amount of movable metal in the ARC demonstration, we decided on an optical, rather than electromagnetic, tracking technology: an Origin Instruments DynaSight optical radar tracker. The DynaSight tracks three LEDs on the head-worn display and a fourth LED on the hand-held barcode reader.

The touring machine uses a Trimble DSM GPS receiver to determine the position of its antenna, which is located on the backpack above the user's head. We subscribe to a differential correction service provided by Differential Corrections Inc., which allows us to achieve one-meter accuracy. We are currently replacing this GPS system with an Ashtech GG Surveyor GPS receiver, which provides real-time centimeter-level position fixes. Orientation tracking is accomplished using the head-worn display's built-in magnetometer, which senses the earth's magnetic field to determine head yaw, and a two-axis inclinometer, which uses gravity to detect head pitch and roll.

Power, Network, and Peripherals. With the exception of the computers, each of the touring machine's other hardware components has relatively modest power requirements of under 10 watts each. We run them all using an NRG Power-MAX NiCad rechargeable battery belt. To communicate with the rest of our infrastructure the touring machine uses Lucent WaveLan spread-spectrum 2Mbit/sec radio modems in both the backpack and hand-held PCs, which operate with a campus network of base stations. In contrast, ARC's machines share a hardwired ethernet. Validation of part identity and position in ARC is accomplished with a PSC Inc. QuickScan barcode scanner that is position-tracked through a single attached DynaSight LED.

20.5.2 Software

Infrastructure. We use COTERIE [24], a system that provides language-level support for distributed virtual environments. COTERIE is based on the distributed data-object paradigm for distributed shared memory. Any data object in COTERIE can be declared to be a shared object that either exists in one process, and is accessed via remote-method invocation, or is replicated fully in any process that is interested in it. The replicated shared objects support asynchronous data propagation with atomic serializable updates, and asynchronous notification of updates. COTERIE runs on Windows NT/95, Solaris, and IRIX, and includes the standard services needed for building virtual environment applications, including support for assorted trackers. It is built on top of Modula-3 [25] and Repo [26], which is our extended variant of the lexically scoped interpreted language Obliq [27].

Graphics package. We use Repo-3D [28], a version of Obliq-3D [29], a scene-graph–based 3D graphics package, which we have modified to provide additional features needed for virtual environment applications and to achieve better performance.

Operating systems. We run Windows NT on the ARC worker's computer and the touring machine Fieldworks to benefit from its support for multitasking and assorted commercial peripherals. We run Windows 95 on the Amity because it does not support Windows NT.

Web browser. Information on the hand-held computer is currently presented entirely through a web browser. We selected Netscape because of its popularity within our university and the ease with which we can control it from another application. To obtain increased performance, we constructed a proxy server that caches pages locally across invocations. This has also been helpful during radio network downtime and for operation in areas without network coverage.

Application software. The original version of ARC was written in several hundred lines of Repo, and supported a single user: the worker. When we decided to add an additional overview display, implemented in a separate program, we needed it to share the task state with the worker's display. Therefore, we modified the ARC prototype to move its single state variable (representing the current task step) into a replicated object, and exported this variable to the network. We imported this variable into our overview display program, and allowed both programs to change the construction step. However, we noticed that this did not give us all the information the overview display needed, especially about when the worker performed incorrect actions. To distribute this information, we added routines to the replicated object that are called when various interesting conditions are noticed, such as the task being completed, the worker scanning the wrong part, and the worker scanning the right part in the wrong location. Note that none of this information is "typical" distributed virtual environment data that would be supported by a distributed VE toolkit, but Repo allows us to distribute the information and react to changes in the various programs in a few lines of code that took a few minutes to write.

The touring machine prototype comprises two applications, one running on each machine, implemented in approximately 3600 lines of commented Repo code. The tour application running on the backpack PC is responsible for generating the graphics and presenting it on the head-worn display. The application running on the hand-held PC is a custom HTTP server in charge of generating web pages on the fly and also accessing and caching external web pages by means of a proxy component.

One of the main reasons that we run our own HTTP server on the hand-held display is that it gives us the opportunity to react freely to user input from the web browser. For example, when a URL is selected on the hand-held display, the HTTP server can call a network object method that selects corresponding graphical items on the head-worn display. Thus data selection works in both directions: from the backpack PC to the hand-held PC (by launching relevant URLs from the head-worn display's menus) and vice versa (selecting buildings, departments, etc. on the

head-worn display from a link on the hand-held's browser).

The HTTP server is initialized by the tour application running on the backpack PC. Each piece of information (buildings, departments, their whereabouts, and assorted URLs) in the tour data on the backpack PC is sent to the hand-held PC's HTTP server with an accompanying procedure closure. The closure executes a procedure on the backpack PC when the corresponding link is selected on the web browser. This makes it possible for the hand-held display to control the head-worn display, as described in Section 20.4.

The web browser on the hand-held PC is a totally separate process. It can be pointed at URLs from within the HTTP server, which we currently accomplish by forking off a separate URL pusher process. The web browser then issues a request back to the HTTP server to obtain either a locally generated, cached external, or uncached external HTML document.

The tour application continuously receives input from the GPS position tracker and the orientation tracker. It also takes user input from the trackpad that is physically attached to the back of the hand-held PC. Based on this input and a database of information about campus buildings, it generates the graphics that are overlaid on the real world by the head-worn display.

20.6 Conclusions and Future Work

We have described two prototype applications that explore approaches to the design of mobile augmented-reality systems. Many hardware issues must be resolved for commercial versions of such systems to become practical, including significant improvements in the quality of tracking and display technologies. We are working on several extensions to our work.

Overlaying virtual objects on the real world can be potentially confusing if they interfere with the user's view of the real world and of each other. For example, even the relatively sparse overlaid graphics of Figures 20.5 and 20.6 evidence problems caused by self-occlusion. We are currently incorporating the Snap-Together Math constraint-based toolkit [30] into our system to explore how automated satisfaction of geometric constraints among objects could help maintain display quality for a mobile user.

We are also working with colleagues in Columbia's Graduate School of Journalism to explore the touring machine's potential for developing and presenting situated 3D news stories that are embedded in the actual locations in which they occurred. Here, we are experimenting with the use of our system as a "mobile journalist's workstation" that presents images and audio on the head-worn display, coordinated with web pages and videos on the hand-held computer. Our first story is a documentary that describes Columbia's 1968 student revolt, as shown in Figure 20.8. This application raises many interesting issues about media coordination, storytelling, and multimedia authoring.

Figure 20.8 Mobile Journalist's Workstation. An image from a 3D news story about Columbia's 1968 student revolt. Protestors outside Hamilton Hall are shown in context of that building.

Acknowledgments

Tony Webster and his Building Technology Lab in Columbia's Graduate School of Architecture, helped develop the design for the ARC and Touring Machine projects. John Pavlik, and students at the New Media Center in Columbia's Graduate School of Journalism, helped develop the concept for the Mobile Journalist's Workstation. Xinshi Sha assisted in developing COTERIE, and Ruigang Yang created Windows 95 utilities. Christina Vernon and Alex Klevitsky helped create the campus databases. Rod Freeman, Jenny Wu, and Melvin Lew helped implement ARC. Jim Foley of MERL, a Mitsubishi Electric Research Laboratory, generously provided a Mitsubishi Amity and Reuven Koblock of Mitsubishi Electric ITA Horizon Systems Laboratory assisted us with the Amity; Marc Najork of DEC's System Research Center and Bill Kalsow of Critical Mass, Inc., assisted during the development of COTERIE.

This work was supported in part by the Office of Naval Research under Contracts N00014-94-1-0564 and N00014-97-1-0838, the Columbia Center for Telecommunications Research under NSF Grant ECD-88-11111, NSF Grant CDA-92-23009, NSF Gateway Engineering Coalition under NSF Grant EEC-9444246; a Columbia University Provost's Strategic Initiative Fund Award, and gifts from Microsoft, Mitsubishi, and Starnet International, Inc.

References

[1] R. Azuma: "A survey of augmented reality," *Presence*, vol.6, no.4, pp.355–385, Aug. 1997.

[2] T. Starner: "Wearable computing and context awareness," PhD thesis, Program in Media Arts and Sciences, MIT, Cambridge, MA, Feb. 1999.

[3] S. Mann: "Wearable computing: A first step toward personal imaging," *Computer*, vol.30, no.2, Feb. 1997.

[4] T. Furness: "The super cockpit and its human factors challenges," *Proc. Human Factors Society 30th Annual Meeting*, pp.48–52, Santa Monica, CA, 1986.

[5] A. State, M. Livingston, W. Garrett, G. Hirota, M. Whitton, E. Pisano, and H. Fuchs: "Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies," *Proc. SIGGRAPH '96*, pp.439–446, New Orleans, LA, Aug. 4–9, 1996.

[6] S. Feiner, A. Webster, T. Krueger, B. MacIntyre, and E. Keller: "Architectural anatomy," *Presence*, vol.4, no.3, pp.318–325, Summer, 1995.

[7] S. Feiner, B. MacIntyre, and D. Seligmann: "Knowledge-based augmented reality," *Comm. ACM*, vol.36, no.7, pp.52–62, Jul. 1993.

[8] T. Caudell and D. Mizell: "Augmented reality: An application of heads-up display technology to manual manufacturing processes," *Proc. Hawaii Int'l. Conf. on Sys. Sci*, pp.659–669, Hawaii, Jan. 1992.

[9] J. Rekimoto and K. Nagao: "The world through the computer: Computer augmented interaction with real world environments," *Proc. UIST '95*, pp.29–36, Nov. 14–17, 1995.

[10] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, and H. Fuchs: "A demonstrated optical tracker with scalable work area for head-mounted display systems," *Computer Graphics (1992 Symp. on Interactive 3D Graphics)*, vol.25, pp.43–52, Mar. 1992.

[11] A. Janin, D. Mizell, and T. Caudell: "Calibration of head-mounted displays for augmented reality applications," *Proc. VRAIS '93*, pp.246–255, Seattle, WA, Sep.18–22, 1993.

[12] R. Azuma and G. Bishop: "Improving static and dynamic registration in an optical see-through HMD," *Proc. SIGGRAPH '94*, pp.197–204, Orlando, FL, Jul.24–29, 1994.

[13] A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston: "Superior augmented reality registration by integrating landmark tracking and magnetic tracking" *Proc. SIGGRAPH '96*, pp.429–438, New Orleans, LA, Aug.4–9, 1996.

[14] J. Loomis, R. Golledge, R. Klatzky, J. Speigle, and J. Tietz: "Personal guidance system for the visually impaired," *Proc. 1st Ann. Int'l. ACM/SIGCAPH Conf. on Assistive Technology*, pp.85–90, Marina del Rey, CA, Oct. 31–Nov. 1, 1994.

[15] H. Petrie, V. Johnson, T. Strothotte, A. Raab, S. Fritz, and R. Michel: "MoBIC: Designing a travel aid for blind and elderly people," *J. Navigation*, vol.49, no.1, pp.45–52, 1996.

[16] A. Smailagic and D. Siewiorek: "The CMU mobile computers: A new generation of computer systems," *Proc. COMPCON '94*, Feb. 1994.

[17] S. Long, D. Aust, G. Abowd, and C. Atkeson: "Cyberguide: Prototyping context-aware mobile applications," *CHI '96 Conf. Companion*, pp.293–294, Apr. 1996.

[18] P. Dykstra: "X11 in virtual environments," *Proc. IEEE 1993 Symp. on Research Frontiers in Virtual Reality*, pp.118–119, San Jose, CA, Oct. 25–26, 1993.

[19] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon: "Windows on the world: 2D windows for 3D augmented reality," *Proc. UIST '93 (ACM Symp. on User Interface Software and Technology)*, pp.145–155, Atlanta, GA, Nov. 3–5, 1993.

[20] I. Angus and H. Sowizral: "VRMosaic: WEB access from within a virtual environment," *Proc. IEEE Information Visualization '95*, pp.59–64, IEEE Computer Society Press, Oct. 30–31, 1995.

[21] S. Feiner and A. Shamash: "Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers," *Proc. UIST '91 (ACM Symp. on User Interface Software and Technology)*, pp.9–17, Hilton Head, SC, Nov. 11–13, 1991.

[22] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger: "Augmented reality in architectural construction, inspection and renovation," *Proc. ASCE 3rd Congress on Computing in Civil Engineering*, pp.913–919, Anaheim, CA, Jun. 17–19, 1996.

[23] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster: "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," *Personal Technologies*, vol.1, no.4, pp.208–217, 1997.

[24] B. MacIntyre and S. Feiner: "Language-level support for exploratory programming of distributed virtual environments," *Proc. UIST '96*, pp.83–94, Seattle, WA, Nov. 6–8, 1996.

[25] S. Harbison: " *Modula-3*," Prentice-Hall, 1992.

[26] B. MacIntyre: "Repo: Obiq with replicated objects. Programmer's guide and reference manual," Technical Report CUCS-023-97, Columbia University Dept. of CS, Aug. 1997.

[27] L. Cardelli: "A language with distributed scope," *Computing Systems*, vol.8, no.1, pp.27–59, Jan. 1995.

[28] B. MacIntyre and S. Feiner: "A distributed 3D graphics library," *Proc. SIGGRAPH '98*, pp.361–370, Orlando, FL, Jul. 19–24, 1998.

[29] M. Najork and M. Brown: "Obliq-3D: A high-level, fast-turnaround 3D animation system," *IEEE Trans. on Visualization and Computer Graphics*, vol.1 no.2, pp.175–145, Jun. 1995.

[30] M. Gleicher and A. Witkin: "Supporting numerical computations in interactive contexts," *Proc. Graphics Interface '93*, pp.138–146, Toronto, Ontario, Canada, Canadian Information Processing Society, May 1993.