Multimedia Systems (1996) 4: 250-268

Multimedia Systems

Future Multimedia User Interfaces

Blair MacIntyre, Steven Feiner
Columbia University
{bm, feiner}@cs.columbia.edu

No matter how powerful or elegant the technologies underlying multimedia computing, it is the user interface that ultimately determines how these systems will be used. We provide an overview of two emerging areas of user-interface research that will play major roles in future multimedia systems: virtual environments and ubiquitous computing. We discuss what they are and why they are important, and describe their relationship to current multimedia systems.

While mouse and window-based user interfaces provide access to 2D graphics, video, and sound, virtual environments not only involve interactive 3D graphics, but further address all our senses through the use of spatial audio, haptics, and other modalities. While it is still a research goal to provide high-throughput video on demand, ubiquitous computing will require that multimedia network infrastructure also support extremely low-latency interaction among large numbers of mobile wireless users. To arrive at an understanding of the needs, capabilities, and potential of these new paradigms, we introduce their terminology and technology, and explain the basic system architectures being explored, leading up to a discussion of key current research issues and future directions.

1 Introduction

A glance at the proceedings of recent conferences on human-computer interaction reveals that multimedia is one of the most active areas of research in the user-interface community. Most of this work takes the current desktop computing environment as a given: a workstation with one or more video displays for output, a keyboard and mouse for input, and increasingly, high-quality audio. Our goal here is to examine a companion set of research areas that go beyond this existing hardware technology to ask how people will interact with future computers that will be quite different physically from the ones we use now.

The place computers currently occupy in our lives is a function of their capabilities, size, expense, and networking technology. High-quality displays are extremely large and expensive, and real-time high-performance 3D graphics hardware, although now available on PCs, still remains a relatively high-end option. Handwriting and speech recognition are tantalizingly close, yet still not mature. With the exception of a few low-bandwidth wireless networking technologies, connected computers are usually stationary computers. Because a computer is a significant investment, people generally need to justify owning one and are unlikely to own many. This has led to the mentality of having a single "personal" computer, perhaps supplemented by a lower-powered portable lap-top or hand-held computer. Given the capabilities and price of current commodity computers, this mindset is justified.

However, this situation will soon change. Two major paradigms that will eventually redefine how we interact with computers are *virtual environments* and *ubiquitous computing*. *Virtual environments* are synthesized worlds created by coupling 3D interaction devices and displays with powerful multimedia computers. *Ubiquitous computing* is a term coined by Mark Weiser [Weiser 1991] to describe a future in which we are surrounded in our everyday life by a multitude of computers so numerous, yet so unremarkable, that we barely know they are there.

In this article, we examine some of the work that has been done in these two fields and explore where they are heading. First, we review their often-confusing terminology and provide a brief historical overview. Since both fields rely largely on relatively unusual, and largely immature, hardware technologies, we next provide a high-level introduction to important hardware issues. This is followed by a description of the key approaches to system architecture used by current researchers. We then build on the background provided by these sections to lay out a set of current research issues and directions for future work. Throughout, we attempt to emphasize the many ways in which virtual environments and ubiquitous computing can complement each other, creating an exciting new form of multimedia computing that is far more powerful than either approach would make possible alone.

e-mail: bm@cs.columbia.edu

1.1 Terminology

Many terms are currently in use to refer to the concepts discussed in this article. To begin with, we prefer the term *virtual worlds* or *virtual environments* to the more commonly used *virtual reality*. The term *virtual reality* promises far more than our technology can currently deliver and has been so abused by the media that it means something different to almost everyone. It has been variously used to describe user interfaces ranging from synthesized physical environments presented on head-mounted displays (HMDs), to ordinary 3D graphics displayed on conventional CRTs, to text-based multi-user games. We use virtual worlds or virtual environments to refer in general to computer-based systems that are 3D rather than 2D, interactive as opposed to passive, and which use one or more devices that attempt to provide a sense of spatial “presence” to the user, be it visual, auditory, or tactile. Among these devices are head-tracked displays and stereo displays (both visual and audio), hand trackers, and haptic displays, all of which we describe later.

Artificial reality is a term coined by Myron Krueger [Krueger 1991]. While he intended it to describe virtual environments in general, it is now popularly used to refer to unencumbered virtual environments that do not require that the user wear special hardware to experience and control a virtual environment. Krueger and others have built systems based on this paradigm, using projection displays and video-based tracking [Krueger 1993; Vincent and Wyshynski 1993].

Augmented reality refers to the use of see-through displays to overlay computer-generated graphics on the user’s view of the real world. Current systems use see-through HMDs [Bajura et al. 1992; Feiner et al. 1993b] but there is no reason that hand-held displays could not also be used. In either case, the user’s head must be tracked to enable the system to render accurate graphics that register properly with the real world.

Telepresence refers to the sense of being someplace other than where you are. One precursor to computer-based virtual environments used an HMD that was coupled to a pair of cameras at a remote location. When the user’s head moved, the remote cameras moved analogously, giving a strong sense of telepresence. (Videoconferencing systems are often claimed to provide a partial feeling of telepresence).

Teleoperation systems allow the user to perform some task at a distance, typically by manipulating a master controller that causes a remote slave effector to mimic the master controller’s movements. The slave effector often provides signals back to the master for force feedback. *Telerobotics* systems control robots remotely.

Ubiquitous computing refers to an environment in which a large number of computers, ranging from hundreds of palm-sized to a few wall-sized units will be seamlessly integrated into our immediate surroundings, connected by wireless networks. When this happens, computers will be so commonplace that we will be no more aware of them than we are of any other tools we use to accomplish day-to-day tasks. Weiser draws an analogy to the electric motor: at one time, motors were major investments, while today, they are small and cheap. We are not consciously aware of all the motors around us. For example, there are dozens of electric motors embedded in a car, but we do not normally think of them as we operate the electric windows, locks, wipers, or ignition.

In a ubiquitous computing environment, people as well as computers will be tracked as they move about. Using this information, a computer would adapt transparently to the preferences of the person using it; the notion of a “personal” computer would become superfluous, as all computers would be aware of a user’s preferences and respond in a predictable way. Unlike most other tools, computers currently become the central focus of a task, instead of simply assisting with it. In a ubiquitous computing environment, computers stay in the background and allow people to concentrate on the task itself.

Ubiquitous computing, with its promise of small, plentiful computers, has also been associated with several other terms. Over the past few years, small hand-held computers known somewhat ambitiously as *personal digital assistants* (PDAs) have begun to appear on the market. Current systems offer little more than calendar, calculator, notepad, address-book, and phone/fax/email functions. However, the popular image of where these devices are headed is exemplified by Apple’s hypothetical Knowledge Navigator [Dubberly and Mitch 1987], a device that knows intimate details about its owner, and can use this information to intelligently assist the owner in managing their daily life.

One of the enabling software technologies for future PDAs will be *agents*. Agents are small semi-autonomous programs that can perform specific activities on behalf of a computer user or even another computer program. Thanks in part to the popularity of the videotape that first presented the Knowledge Navigator and earlier science fiction that inspired it, agents are often depicted as having a human persona that implies the sophisticated, broad ranging skills of a personal assistant or butler.

Finally, *computer augmented environments* is a term used by Pierre Wellner, Wendy Mackay, and Rich Gold [Wellner et al. 1993] to refer to any technology that adds computational power to the real world. This is an intentionally broad concept, ranging from the physically embedded machines of ubiquitous computing to the overlaid graphics of a projection display or a see-through head-worn display.

1.2 Virtual Environments, Ubiquitous Computing, and Multimedia

Most proponents of ubiquitous computing agree with Weiser that the “most diametrically opposed [environment] to our vision is the notion of virtual reality, which attempts to make a world inside the computer” [Weiser 1991]. While he notes several applications that will benefit strongly from virtual reality, such as games and virtual walk-throughs of inaccessible environments, he argues that “virtual reality” is not appropriate for day-to-day use by the average user. Here, Weiser is using the term to refer to virtual environments that use today’s cumbersome, low-resolution, opaque HMDs and gloves to shut the user off from friends, colleagues, and all the richness of the real world.

We agree that even when far better versions of these technologies become available, self-contained virtual environments may still have relatively limited uses. In contrast, however, we believe that virtual environments in general will play a major role in the future of ubiquitous computing. This can be possible by extending the vision of ubiquitous computing to include descendants of the technologies used in current augmented reality research. Many detractors of HMDs point out that people will not want to wear bulky hardware on a daily basis. But, treating today’s user-interface hardware as if it would not improve is like predicting that people will never want to carry around high-performance computers because they will weigh too much. Just as surely as computers will get considerably smaller, today’s bulky HMDs will be replaced by lightweight, comfortable, inexpensive, and attractive glasses-like displays. In many ways, the see-through display has the potential to be the Sony Walkman of the early 21st century.

Both virtual environments and ubiquitous computing are inherently multimedia paradigms. Virtual environments can involve all the senses: visual, auditory and tactile components can be combined to create convincing virtual worlds. Furthermore, these worlds involve not only synthesized material (e.g., computer graphics) and recorded material (e.g., video), but, in the case of augmented reality, the surrounding real world itself.

Ubiquitous computing also naturally integrates all the traditional digital media: text, audio, synthesized graphics and video. However, because of the large number of displays available per person, the potential for intensive use of multimedia is significantly better than with current technologies. An entire display can be devoted to a video connection or still image and left that way for long periods of time. Audio will become more important as the proliferation of displays increases the need to use clearly identifiable sounds to call attention to important information and minimize confusion. Moreover, many of the issues that need to be addressed in conventional multimedia systems, such as network and operating system support, are vital to the success of virtual environments and ubiquitous computing.

1.3 Historical Overview

In this section we briefly highlight some of the important milestones in the history of virtual environments and ubiquitous computing. For a more thorough history of virtual environments, see [Kalawski 1993]. For an introduction to ubiquitous computing, see [Weiser 1991] or [Weiser 1993a].

It is interesting that the first examples of virtual environments did not use computers. Morton Heilig developed a machine called the Sensorama [Heilig 1962] whose demo took the user on a virtual motorcycle ride through New York that addressed many of the senses simultaneously: visual (movie images), auditory (stereo sound), haptic (vibrations and wind) and olfactory (city smells). He also invented an immersive stereoscopic HMD for visual, auditory and olfactory stimulation [Heilig 1960]. Early flight simulators created virtual environments without the aid of computers by using user-controlled live video, generated by shooting model boards with servo-driven TV cameras [Schachter 1983].

In 1965, Ivan Sutherland gave a talk at the IFIP conference in which he described his notion of the “ultimate display,” a room in which synthesized objects would be as convincing to all our senses as real objects [Sutherland 1965]. Over the next several years, he developed the first virtual environment with a head-tracked HMD. Sutherland’s system presented 3D vector graphics on small CRTs worn at the sides of the user’s head. The optical path was folded using half-silvered mirror beamsplitters, also making it the first augmented reality system.

Inspired by Sutherland’s talk on the ultimate display, Fred Brooks began work on haptic displays that present force and torque, combined with 3D graphics, many of which have been tested in prototype biomolecular modeling applications [Brooks 1988].

In the early 1970's, Michael Noll built a prototype computer-based 3D force feedback device. It used potentiometers to sense the position of a small knob on a vertical shaft and computer-driven motors to control the force exerted on the knob [Noll 1972]. The user could hold the knob and move it within a 10" cube to explore a simulated object, whose surfaces could be varied dynamically from hard to rubbery.

Starting in the 1970's, Myron Krueger built a number of systems that carefully avoided technologies that encumbered the user with bulky wearable displays and sensors, opting instead for display and tracking approaches that traded off physical immersion for comfort and appeal [Krueger 1991].

Researchers at NASA Ames [Fisher et al. 1986] developed many of the key ideas in modern virtual environments in the 1980's, including color stereo HMDs and spatial sound [Wenzel and Foster 1990]. In the mid 1980's, VPL Research created the DataGlove and popularized the HMD, both of which gained considerable media attention. In the succeeding years, dozens of virtual environments development and consulting firms, and many research labs, have been formed.

The roots of ubiquitous computing lie in the history of personal computing, beginning with the pioneering work of Alan Kay and his colleagues at Xerox PARC in the 1970's. They proposed the Dynabook, a powerful book-sized computer intended for use by anyone, from children to adults [Kay 1977]

As interest in PDAs grew in the mid-1980's, many researchers proposed that PDAs would one day serve as intelligent agents that would perform a variety of tasks, from searching databases to buying airline tickets [Dubberly and Mitch 1987]. Not everyone was enamored by the prospect of a world filled with anthropomorphic assistants. Ubiquitous computing research began at Xerox PARC in the late 1980's to explore an alternative view of the place future computers would occupy in everyday life [Weiser 1993a].

2 Hardware Technology

In this section we provide a high-level overview of the hardware used in current virtual environment and ubiquitous computing systems: spatial trackers, and graphics, audio, and haptic displays and interaction devices. Rather than trying to offer comprehensive coverage, we include just enough detail to allow the reader to understand the capabilities available at the user interface.

2.1 Spatial Trackers

A variety of spatial tracking systems exist that measure and report information about position, orientation, acceleration, or joint angles. For example, six-degree-of-freedom (6DOF) sensors provide both 3D position and 3D orientation information. Two 6DOF tracker technologies are currently in popular use: electromagnetic and ultrasonic.

(a) (b)

Figure 1: Two popular spatial tracking technologies. (a) Electromagnetic (Ascension Technologies Flock of Birds, courtesy Ascension Technologies, Inc.), (b) Ultrasonic (Logitech Flying Mouse, courtesy Logitech Inc.)

Electromagnetic trackers, produced by Polhemus and Ascension, use a stationary transmitter that contains three mutually orthogonal coils, embedded in a cube that ranges from a few inches to a foot on a side. The coils are pulsed in sequence with an electrical signal so that each produces a magnetic field. A smaller receiver, a cube about an inch on a side, contains three coils that sense the transmitter's magnetic fields, enabling the system to determine the receiver's position and orientation. Depending on the system, the receivers can be tracked to a distance of about three to twenty feet from the transmitter. Electromagnetic trackers do not require a direct line of sight between tracker and receiver, but they are sensitive to metallic objects and magnetic fields (Figure 1a).

Ultrasonic trackers, such as the Logitech tracker, use a stationary transmitter that contains three small sources (speakers) arranged in a triangle, that emit a series of ultrasonic pulses. The receiver contains three small microphones that sense the pulses. Based on the delay between when the pulses were transmitted and received, the system can determine the distance between each speaker and each microphone, and hence the relative position of each microphone. From these three positions, the orientation of the receiver can be recovered. Ultrasonic trackers are not affected by the presence of metal or magnetic fields, but are sensitive to noise and reflections, and require a direct line of sight between transmitter and receiver (Figure 1b).

The accuracy and precision of both electromagnetic and ultrasonic trackers leave much to be desired, though they can measure position within a small fraction of an inch and orientation within a small fraction of degree. Both technologies also currently require a cable from the receiver to an electronics box.

Optical tracking systems use video cameras to track objects. Two approaches to optical tracking exist, which are often referred to as *outside-in* and *inside-out*. *Outside-in* systems position cameras in the world that can see the objects they are supposed to be tracking. By using two or more cameras, accurate 6DOF tracking is possible under controlled conditions. For example, some systems require that the objects to be tracked have visual aids attached to them, such as LEDs (e.g., [Gaertner 1992]) or markings. *Inside-out* systems attach cameras to the objects to be tracked. By watching how the world moves around the camera, the position and orientation of the cameras (and thus the object to which they are attached) can be determined. As with outside-in tracking, existing systems do not work in general environments. For example, the University of North Carolina at Chapel Hill (UNC) room-sized inside-out tracker uses a ceiling whose tiles contain a dense grid of LEDs [Fuchs et al. 1990].

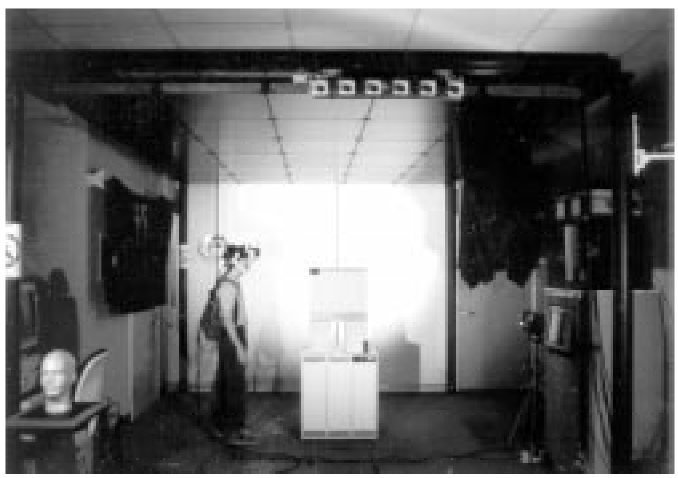


Figure 2: The University of North Carolina at Chapel Hill optical tracking system. The ceiling above the user is instrumented with a dense grid of LEDs. The system computes the users head position based on the set of LEDs visible to each of the cameras mounted on his head. (Courtesy of Ronald Azuma, University of North Carolina at Chapel Hill)

Several other technologies for tracking position or orientation are also being explored. These include gyroscopes, magnetic compasses, inclinometers, and accelerometers, which can be used to create self-contained, wearable systems that do not require an external transmitter or receiver.

Coarser-grained position-only tracking can be accomplished worldwide through the *global positioning system* (GPS). GPS receivers receive signals from positioning satellites maintained by the US government[Getting 1993] and use these signals to determine the absolute position of the GPS receiver's antenna on the earth. Accuracy is currently quite coarse, reporting positional information that can be off by tens of meters. Much better accuracy can be obtained by positioning an additional receiver and transmitter in a precisely known position and having it broadcast corrections based on the difference between its computed and known positions. This technique is called *differential GPS* and can result in accuracies of less than one meter. An even more accurate kind of GPS system compares the phase of GPS signals at a stationary receiver with the phase of signals received by the moving receiver, and can achieve centimeter-level accuracy in real time.

One local, coarse-grained tracking system is based on wearable *active badges* [Want et al. 1992]. An active bade contains a small infrared transmitter that periodically transmits a unique identifier associated with the owner of the badge. The transmissions are detected at a short range by receivers placed in the rooms and hallways of a building being monitored. This makes it possible to tell which room or section of hallway is occupied by the person wearing the badge. Because IR is fairly directional and requires a relatively clear line-of-sight for reception, users desiring privacy can simply put the badge in a pocket, preventing the receivers from picking up the transmitted signal.

Instrumented gloves, such as the VPL DataGlove [Zimmerman et al. 1987], Virtex CyberGlove, and Exos Dexterous Hand Master, provide information about finger joint angles. The DataGlove accomplishes this with a set of ten fiber-optic cables, one for each of the first two joints of the hand's five fingers. Each cable is run in a loop along the back of its finger, and damaged where it passed over its assigned joint. A tiny LED at one end of the cable and a photosensor at the other measure the amount of light that passes through the cable. The more the finger is bent at the joint being measured by the cable, the more light is lost, providing an indirect measure of the joint angle. In addition, the position and orientation of one part of the glove is measured by an electromagnetic 6DOF sensor. The technologies used to build gloves can also be applied to track more of the user's body, as part of a "data suit."

A low-cost system for finger acceleration tracking has been prototyped which uses accelerometers on each finger [Fukumoto and Suenaga 1994]. The developers propose using their system as a "full-time wearable interface" for which any solid surface could serve as a chord keyboard.

Camera-based approaches make it possible to track hands and bodies without requiring the user to wear special apparatus. While general computer vision-based solutions will eventually predominate, it is currently possible to process the user's silhouette to recognize coarse hand and body poses in real time [Krueger 1991; Vincent and Wyshynski 1993; Wellner 1993].

Eye-tracking technology has been under development for over a decade. In one approach, an infrared light illuminates the user's eye and an infrared-sensitive camera tracks the dark pupil, which reflects much less of the infrared than the rest of the eye. This system can measure eye orientation within less than a degree, but requires that the user wear a head-mounted camera or be positioned in front of a stationary camera. An alternative approach monitors the electrical impulses to the muscles around the eye; to date, these systems can only discriminate between a few very large areas over the user's entire visual field. In all cases, eye-tracking systems must contend with involuntary eye movements (saccades) that all users make, and the fact that the eye motion is already part of our normal activity: trying to use the eyes to point can result in the so-called "Midas touch" phenomenon, in which a careless look has unfortunate consequences.

2.2 Graphics Displays and Input

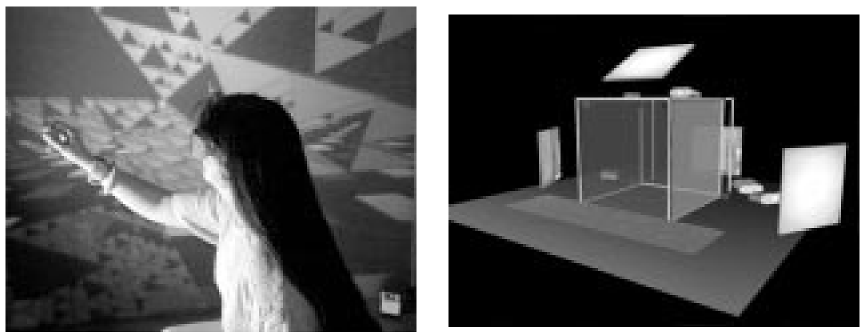
The graphics displays used to create virtual environments are often categorized as *immersive* displays, which surround the viewer with graphics, and *non-immersive* displays, which do not. There are two basic kinds of immersive displays, both of which provide a sense of presence by filling up a relatively large amount of the user's field of view: head-mounted displays (HMDs) and surround displays.

An HMD is actually mounted on the user's head, and contains one or two displays viewed through special lenses that are designed to allow the user to focus on the displays comfortably (see Figure 3). Two displays allow stereo viewing, while one display may be designed to be viewed by one or both eyes. HMDs are usually used in conjunction with a 6DOF tracker to allow the material being viewed to depend on the position and orientation of the user's head. This makes it possible to build up a virtual environment that appears to surround the user as he or she moves and looks about.

In contrast, surround display systems surround the user, not just his or her head. This can be accomplished by building a room whose walls, and possibly floor and ceiling, are themselves displays. One approach is to make each such surface into a rear-projection video screen [Cruz-Neira et al. 1993; Deering 1993], as is done in the CAVE, shown in Figure 4. To provide stereoscopy and sensitivity to head position, these systems are often used with a stereoscopic display system and a head position tracker. One current stereo display technology that is especially well suited for this application relies on eyewear with liquid crystal lenses whose opacity can be controlled by impressing a voltage across them. Left and right eye images are alternately displayed on each screen at double the standard frame rate. The lenses are controlled in synchronization with the display, so that the left lens is transparent only when the left eye's image is displayed, and the right lens is transparent only when the right eye's image is displayed.

Another distinction is between *opaque* and *see-through* displays. Opaque displays block the users view of the real world and present the user with a synthesized visual environment. In contrast, see-through displays overlay

Figure 3: A see-through head-mounted display. (Courtesy of Virtual I/O Inc.)



(a) (b)

Figure 4: The CAVE. (a) The user wears liquid crystal stereo glasses to perceive 3D effects. (b) Images are projected onto three walls, which are rear-projection screens, and directly onto the floor. Mirrors are used to fold the optical paths from the video projectors to minimize the amount of space needed outside the CAVE. (Courtesy of Carolina Cruz-Neira, University of Illinois, Chicago)

graphics on the user's view of the real world. Both opaque and see-through displays can be head-mounted, hand-held, or stationary.

Current see-through displays use either optical combiners (essentially, half-silvered mirrors) to overlay graphics onto the world, or videomixing to merge a video image of the world with graphics on an opaque display. The use of optical combiners means that all overlaid graphics will be transparent, but allows the user to see the real world with the full field of view and acuity that their visual system can provide. In contrast, videomixing allows opacity control at the level of an individual pixel. However, it reduces the user's visual experience of the real world to the resolution and field of view of the opaque HMD and introduces additional lag via the video circuitry. Furthermore, unless a videomixed HMD's cameras can be located at positions equivalent to those of the user's eyes, the image of the world will be offset.

Non-immersive displays range from conventional CRTs to wall-sized displays. These are usually used with some combination of head-tracking and stereo, exploiting the same technologies used for surround systems. The terms *fish-tank virtual worlds* or *desktop virtual worlds* are often used for these systems. The first term derives from the illusion that there is a cubic 3D volume inside the CRT "fish tank."

In describing his group's work on ubiquitous computing, Weiser mentions three basic classes of displays: tiny "tabs" that can be held in one's palm (Figure 5a), hand-held "pads" (Figure 5b), and wall-mounted "boards" (Figure 6) [Weiser 1991]. As pointed out before, while ubiquitous computing and virtual environments have sometimes been presented as polar opposites, many of the displays used in virtual environments could also play an important role in ubiquitous computing. For example, Fitzmaurice [Fitzmaurice 1993] describes the Chameleon, a hand-held LCD display, fitted with a 6DOF tracker, and used as a movable window onto a 3D information space. It could be classified as both a ubiquitous computing and a virtual environment device. The techniques used by the Chameleon could be applied quite naturally to a see-through hand-held display to create an even richer information space.

Ubiquitous computing and virtual environments employ a wide range of interaction devices that are as varied as the displays. The three classes of ubiquitous computing devices discussed in [Weiser 1991] all use pens for input

(a) (b)

Figure 5: (a) The PARCTab is an example of a display used in a ubiquitous computing environment. It has a low resolution display, pen interface, three buttons and an infrared link for wireless communication. (Courtesy of Xerox PARC.) (b) The Zenith CruisePad is an example of a hand-held pad sized display. (Courtesy of Zenith Data Systems Corporation.)

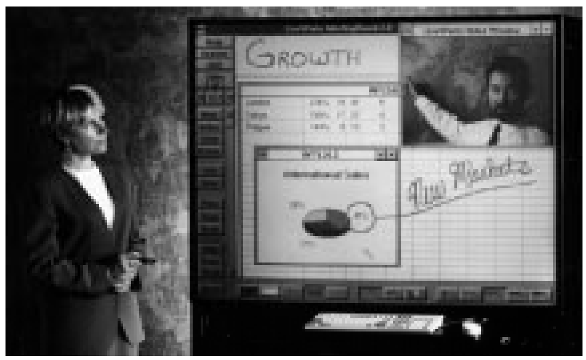


Figure 6: The Xerox Liveboard is an example of a wall-sized display. (Courtesy of Xerox LiveWorks.)

because that seems to be the most suitable input device for hand-held and wall-mounted displays. Of course, traditional keyboard devices are also used, especially for entering large amounts of data. In contrast, virtual environments researchers have explored many different kinds of 3D input devices and techniques. These include simple hand gesture recognition (e.g., using an instrumented glove) and hand-held “interaction controllers,” such as the Digital Image Design Cricket, shown in Figure 7.

Figure 7: The Cricket, a 3D interaction device used in virtual environments. The flat front face of the left Cricket has three microphones for use with a 6DOF ultrasonic tracking system. The array of small circular bumps seen on the handle of the right Cricket provides haptic feedback. (Courtesy of Digital Image Design)

2.3 Audio Displays and Input

Most virtual environments and ubiquitous computing research, like most 2D computing environments, have emphasized the use of graphical displays and input. Nevertheless, many researchers in these areas have been exploring the use of sound. In virtual environments, this work has concentrated mostly on audio output to express properties of virtual objects, and their interactions, that might be difficult to see visually, and on the use of speech input to accompany, disambiguate, or replace gestures. In ubiquitous computing, the emphasis has been on integrating portable computers with wireless telephone services to create personal communicators.

In current multimedia systems, sound output has usually been treated as a companion to graphics presented on a flat display flanked by stereo speakers—video and audio are both wedded to the screen. However, just as head-tracked 3D graphics can create a compelling sense of visual immersion, head-tracked 3D sound [Begault 1994] can immerse the listener in a surrounding audio environment. Similar to the way that the image of an object in three space changes depending on its distance and orientation relative to the viewer's eyes, the sound of an object changes as a function of its distance and orientation relative to the listener's ears. In the case of stereo, left and right ears hear a sound differently, just as left and right eyes see an object differently. 3D sound can be especially useful for helping the listener make sense of what would otherwise be audio clutter. For example, in the well-known "cocktail party effect" listeners can often pick out and attend to one conversation from many ongoing ones when listening in stereo, but cannot do so in mono.

There are three main cues responsible for our hearing a sound as coming from a particular direction. Differences in a sound's amplitude and arrival time at the left and right ears are strong directional cues. However, these alone cannot account for our ability to hear sounds as being above or below, or, for that matter, as being outside our heads. For example, a sound source that is equidistant from both ears reaches each at the same time and with equal amplitude, yet it sounds different as it travels in a circle from in front to above, to behind, to below us. This asymmetry is caused by the way in which the asymmetric pinnae (outer ears), and secondarily the head and torso, shape the sound spectra in a way that depends on the direction of the sound source relative to the head. This filtering is called the *head-related transfer function* (*HRTF*). By inserting small microphones in a subject's ears and measuring the waveforms caused by real sound sources located at different positions around the subject, mathematical models of HRTFs can be developed.

A 3D sound spatialization system (e.g., the Convolvotron [Wenzel and Foster 1990]) uses digital signal processing technology to process a sound source in real time. It applies a simplified version of the HRTF, indexed by the desired location of the source relative to each ear. The result is a stereo pair of signals that when presented to the listener's ears will sound like the source emanating from that relative location. Simple systems model the location of sounds in an anechoic environment; more sophisticated ones include the additional effects of absorption and reflection from room surfaces.

2.4 Haptic Displays

Haptic displays address the user's sense of touch and temperature, in addition to the sensations of force and torque. In our interactions with conventional input devices such as keyboards and mice, touch and force have always been important. For example, it is well known that it is difficult to touch type on a membrane keyboard whose keys do not offer the tactile feedback of a traditional keyboard. This kind of passive haptic feedback is a direct result of the way in which the physical keyboard mechanism is constructed and is not controlled by the computer. In contrast, haptic displays are under active computer control.

Building on Noll's work, mentioned in Section 1.3, a number of research systems have provided promising examples of the power of haptic displays. A motor driven arm, held in the user's hand, has been used to allow biologists to experience the molecular forces between simulated molecules [Brooks, Jr. et al. 1990] and chemists to explore the surface of a sample scanned by a remote electron microscope, inspecting and modifying features at the atomic level by controlling the microscope [Taylor, II et al. 1993]. A motor-driven joystick has been used to produce the illusion of feeling patches of textured material [Minsky et al. 1990]. On the other end of the spectrum, a simple tactile display in the form of an array of small computer-controlled pins provides less dramatic tactile feedback to the user of the Cricket (Figure 7).

3 Architectures

The software architectures used in virtual environments and ubiquitous computing have evolved from a different set of goals. Most virtual environment systems have been developed to support a single hardware platform or a small number of tightly coupled platforms, with the intent of creating stand-alone applications. As a result, these systems were originally modelled after traditional interactive programs and evolved to solve the problems particular to virtual environments. Most existing systems ignore issues that would arise if such applications were used on a larger scale, as will be required for realistic virtual worlds. For example, little work has been done toward developing the infrastructure to support virtual environments at the network and operating systems levels.

The work on ubiquitous computing, on the other hand, began with a vision of what multi-user, multi-computer environments would look like in the future. As a result, the prototype systems being built are being designed to address the infrastructural issues.

3.1 Process Models for Virtual Environments

Many early virtual environment applications were simple event-loop–based programs, in which each "interesting event," such as a head or hand movement, was handled sequentially. The problem with this approach is that the frame rate is tightly coupled to the time taken through each iteration of the main event loop: it is the sum of the time required to read all the input devices, process one step of the virtual environment simulation, and generate the next display frame. For any moderately complicated virtual environment, the time taken to process one step of the simulation is nontrivial. For example, a single timestep in a significant scientific visualization could take many seconds or minutes. This is clearly unsatisfactory: for a virtual environment to be convincing, the displays should respond to changes in tracked objects, especially the user's head, at least ten times per second.

Since the simulation determines *what* is displayed and the user's head position determines *where* it is displayed, the obvious solution to this problem is to distribute the system over multiple processes, decoupling the computation of each step of the simulation from the redisplay loop [Shaw et al. 1993]. The result is that the displays are updated as fast as possible when the head position changes. The information being displayed continues to changes only as fast as each step of the simulation, as before.

In many current virtual environment systems, the distributed processes are tightly coupled. One process is the master and is responsible for creating and initializing all other processes. Such systems are neither easily reconfigurable at run-time nor fault tolerant. Furthermore, they are more difficult to construct because the master application must configure and control all of the devices it wants to use. While these problems are not different in kind from those encountered in a shared 2D drawing system, the message traffic demands of a virtual environment are typically greater, in both volume and complexity, because of the larger number of user interaction devices and the data communication needed to support distributed simulations. To alleviate these problems, some toolkits, such as dVS [Grimsdale 1991], VR-DECK [Codella et al. 1993] and DIVE [Carlsson and Hagsand 1993], structure the applications as a large set of asynchronous, event-driven processes. Each process is independent of the others and communication is via a well-defined message protocol.

A more significant problem with existing systems is that they do not scale up gracefully. To build high-fidelity virtual worlds, we will have to distribute the computational load over a significantly larger number of machines than current systems are designed to handle. Even those existing systems that are distributed tend to centralize the computation of each of their components, such as the virtual environment simulation or collision detection between objects, and communicate the results to all interested parties. The fundamental problem here is that centralized algorithms do not scale well to huge numbers of entities, primarily because the bandwidth of the communication media is quickly overwhelmed as larger numbers of machines need to be informed of the results of these computations.

The first system to address the problem of scale was SIMNET, ARPA's SIMulator NETworking project [Katz 1994; Calvin et al. 1993]. SIMNET was developed to allow hundreds or thousands of simulators to be used for training military personal in collaborative tactics. A more recent system, NPSNET [Zyda et al. 1992], uses similar techniques, but runs on graphics workstations, instead of special-purpose hardware. The distributed interactive simulation (DIS) protocol used in SIMNET (and understood by NPSNET) consists of a few dozen network packet types that allow heterogeneous simulators to communicate. A unique feature of the protocol that allows such a large number of participants is the inclusion of velocity and acceleration information in the position messages for each object. Because most participants in the simulation are vehicles which typically travel in a fairly predictable trajectory, the position of an object can be calculated locally by all participants, relieving the sending entity of the need to continually rebroadcast. When the error between the real and calculated position exceeds a certain threshold, new position information can be broadcast. However, such an approach may not be nearly as successful with the hand and head motion of human participants, which do not move quite so predictably.

The WAVES system [Kazman 1993] is an example of a system currently being developed with the goal of dealing with some of the requirements of large scale distribution of virtual environments over communication media of varying bandwidth in a more general fashion than SIMNET or NPSNET.

3.2 Infrastructure for Ubiquity

There are three main components to a ubiquitous computing environment: inexpensive, low-power, mobile computers and displays, a software infrastructure for ubiquitous applications, and high-capacity wireless networks that tie everything together [Weiser 1991]. The network requirements are discussed in Section 4.1.4. Given the current rate of technological advancement, the computing and display requirements are likely to be met within a decade. This leaves the software infrastructure to be developed.

Wireless infrastructures suitable for testing palm-sized and pad-sized displays and schemes for network routing are actively being explored [Schilit et al. 1993a; Ioannidis and Maguire 1993; Weiser 1993a]. More general operating systems issues for palm-sized computers, such as dealing with heterogeneous networks and devices, are discussed in [Theimer et al. 1993].

Other problems include how to deal with the widespread effects of intermittent connectivity in wireless networks, such as handling intermittent connections to fileservers. Many filesystems for mobile computing use transparent local caching to alleviate the problem somewhat. However, because the caching is done transparently, applications have no way to determine which files and operations are actually available. The result is that applications become very unpredictable because the user never knows which operations are possible. Tso investigated one way to deal with this, by exposing, in a structured manner, the caching filesystem to the applications, which could use this information to inform the user what operations and files are available [Goldberg and Tso 1993; Tso 1993].

Another issue is how to inform applications that their environment is changing so they may customize their interface appropriately. For example, an application should be able to be told to “print on the nearest printer” without explicitly naming the printer. One approach to this problem is to modify the semantics of environment variables, introducing the notion of *dynamic environment variables* which automatically inform interested applications when their values change [Schilit et al. 1993b]. The example application simply needs to be configured to print to the “local printer,” and the system will inform the application when the value of the local printer variable changes.

Ubiquitous computing environments also promise increased capabilities from devices other than computers. For example, Elrod has created a responsive office environment that provides users with intelligent control over their surrounding by making automatic adjustments to lighting and heating systems based on the preferences of the room’s occupants [Elrod et al. 1993a; Elrod et al. 1993b]. In addition to convenience, this system also promises to save significant amounts of energy if used on a company-wide scale to turn off equipment in vacant rooms.

3.3 Display Generation Models

The popular image of virtual reality is of immersive, fully synthesized, photorealistic graphics. While current technology is far from providing this, many people feel this is the ultimate goal toward which display generation research should strive. In a sense, they are correct; ideally, such an environment has the potential to provide the visual sense of complete immersion. But for many applications, complete immersion is either unnecessary or inappropriate. Furthermore, the technology to achieve a convincing, fully synthesized virtual environment will not be available in the near future. Here we discuss three display generation models for virtual environments: fully rendered scenes, augmented reality and image processing.

3.3.1 Fully Rendered Scenes

The most straightforward and common approach to creating virtual environments is to render a fully synthesized frame for each user from his or her viewpoint. Given a model of the virtual world and the user’s head position, the parameters of the synthetic cameras associated with the user’s eyes can be calculated, and the scene rendered using traditional computer graphics techniques. For monoscopic displays, the scene is rendered once; for stereoscopic displays, the scene is rendered twice, once from the point of view of each eye.

In an HMD, the small displays are close to the user’s eyes, fixed with respect to the user’s head and eye positions, and constantly moving with respect to the world. In a desktop system, the display is relatively far from the user’s head and fixed with respect to the world, but constantly moving relative to the user’s head and eyes. While these are conceptually major differences from the user’s perspective, from the standpoint of the graphics computations, the only difference between head-tracked displays that are head-mounted or desk-mounted is a small difference in the parameters supplied to the synthetic cameras used to render the graphics.

The key feature of both environments is that *everything* in the virtual world must be rendered, though clever approaches can be used to reduce the number and complexity of objects that must be rendered for each frame. Various techniques, such as texture mapping, can also reduce the number of polygons that must be drawn, but everything the user sees must be created by the system. The obvious disadvantage of this approach is that extremely powerful computers are required, and it is currently not possible to create complex, realistic environments in real time.

3.3.2 Augmented Reality

An alternative to fully synthesized graphics is to render only selected objects that are combined with what the user can already see in the real world. Unlike fully synthesized environments, only those entities that do not exist in the real world need to be rendered. To highlight an object, for example, it is sufficient to draw a wireframe box or circle around it, as shown in Figure 8. Virtual worlds that communicate a wealth of information when they annotate the

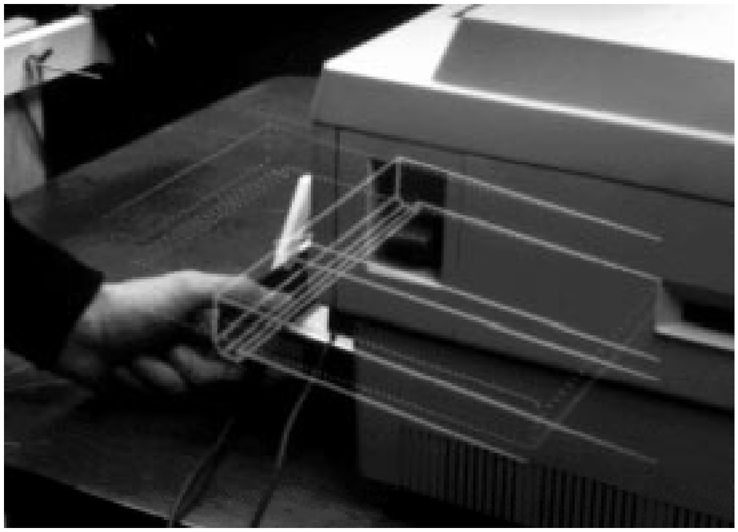


Figure 8: In an augmented reality environment, only the additions to the world need to be rendered. In this example, from the KARMA maintenance assistant [Feiner et al. 1993b], the user is being shown how to pull out a printer's paper tray. The current and desired positions of the paper tray are highlighted using different line styles and a pulsing arrow is drawn to show the user the direction in which the tray should be moved. (Courtesy Feiner, MacIntyre, and Seligmann, Columbia University.)

real world can be created using a few hundred lines or polygons [Feiner et al. 1993b]. Furthermore, the user is not cut off from the richness of the real world: mundane objects such as pens and paper are still accessible, as are other people and computers.

3.3.3 Image Processing

Rather than rendering the entire scene from the viewpoint of each user, we can instead render (or photograph) a 360˚ wide field-of-view image of the scene once from a single position. Image processing techniques can then be used to produce an image rendered from the viewpoint of a given user [Lippman 1980; Regan and Pose 1993]. This works if all objects are distant from the user and if each user is positioned at roughly the same location, although looking in an arbitrary direction. This is a potentially important approach for broadcast video games [Warp 1994].

Nearby objects can be added by rendering them separately for each user, and simply overlaying them on the distant background if they are guaranteed to be closer than the rest of the scene. For example, shared live video imagery of a roller coaster ride could be overlaid with synthesized images of the interior of the roller coaster car, generated for each user. If the synthesized information were generated locally, a shared virtual experience could be broadcast to a group of people.

4 Research Issues and Directions

A number of issues will need to be addressed if virtual environments and ubiquitous computing are to supplant current approaches. The most obvious of these are the need for improved hardware: tracking systems, displays, processors and communication networks need to be substantially improved before virtual environments and ubiquitous computing can hope to gain widespread acceptance. While it may be some time before these technologies mature, there are many other issues, both technical and social, that should be pursued in the mean time. For example, appropriate interaction techniques need to be developed for these new paradigms. Automated assistance for authoring, application of hypermedia and agent technologies, support for collaboration and privacy are all areas ripe for exploration.

4.1 Hardware Requirements

Several important hardware issues are the mobility of computers and the speed of both computers and the wireless networks connecting them. Work needs to be done in four areas: trackers, displays, processors, and networks.

As portable computers become more powerful and wireless networks become faster, the mobility of virtual environment applications is being increasingly limited by tracking and display technologies. Similarly, as the speed of computers and networks increases, the primary limit on system responsiveness becomes tracker lag. For ubiquitous computing, important hardware factors are the physical size of computers and the bandwidth of wireless networks.

In the foreseeable future, processors will continue to get smaller, faster and require less power. The same holds true for traditional display technologies of the sort required for tabs or pads. The bandwidth and speed of wired networks are also increasing. While all these advances are necessary for the future of ubiquitous computing and virtual environments, we will avoid discussing them here except where the specific requirements of ubiquitous computing or virtual environments differ from those of what are currently mainstream applications.

4.1.1 Trackers

Virtual environments and ubiquitous computing currently use completely different tracking systems. Virtual environments require extremely fine-grained tracking, whereas ubiquitous computing uses long range, wireless systems. Since no system delivers all these features, trade-offs must be made: relatively high accuracy, short-range tethered systems versus realtively low accuracy, long-range, wireless systems. As a result, the total range of most tracking systems currently used in virtual environments is comparable to the smallest unit of accuracy typically used in ubiquitous computing! However, it is important to keep in mind that both paradigms can take good advantage of wide-area, high-accuracy systems. For example, experimental ubiquitous computing environments are currently customized based on who is occupying a room. With higher accuracy, the environment could be customized based on *exactly* where everyone is, what display they may be holding, or where they are looking.

The problems with current tracking systems, as discussed in Section 2.1, can be categorized as ones of range, latency, and accuracy. The relative importance of these attributes depends on the specific use of the tracker, be it a 6DOF head tracker for an HMD or a coarse location tracker. Within a virtual environment, the requirements differ depending on whether the system is immersive, desktop-based, or an augmented reality. For all systems, low latency is important, especially when the user's head is being tracked. Azuma [Azuma 1993] points out that the range and accuracy requirements of immersive, completely virtual environments are much less stringent than the requirements of other virtual worlds: it suffices to know only the approximate position and orientation of the user, because the user's visual sense tends to override conflicting signals from their other senses. Furthermore, if the range of the trackers is limited, various techniques, such as "flying," allow the user to navigate within the immersive virtual world without physically moving.

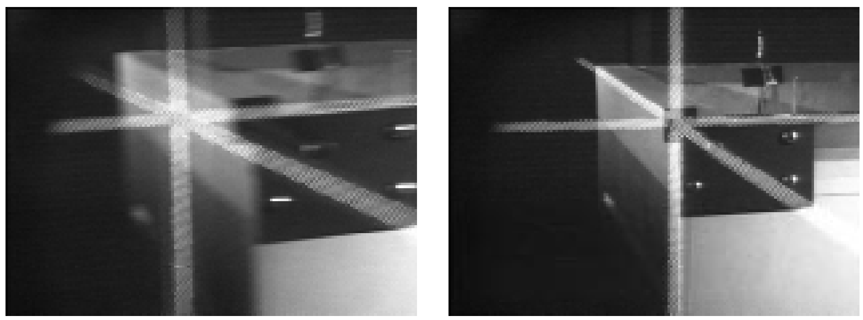
Desktop virtual environment systems do not require significant range because the user must be close enough to use the stationary display; tracking a user's head within a few feet of a hand-held display or within the room containing a wall-mounted display is often sufficient. Augmented reality systems, on the other hand, have the strictest requirements. Extremely high accuracy and low latency are needed to ensure that the virtual images register properly with the real world. Furthermore, just as with ubiquitous computing, there are potential applications that will require tracking on a building-wide, city-wide, or even planetary-wide, scale.

Unfortunately, magnetic and ultrasonic systems do not scale well. Because they use a transmitter as well as a receiver, their range will always be limited by the requirement that both be located within a matter of feet. Objects in the environment can also produce noise (magnetic or ultrasonic) that interferes with the system, requiring lag-inducing filtration.

Some of the most promising research on long range, highly accurate tracking is being done in the experimental inside-out optical tracking project at UNC, mentioned in Section 2.1. This tracking system scales well and exhibits uniform high accuracy and low latency over its entire range [Azuma and Bishop 1994]. The major drawbacks of the current system are its size and weight, the degree to which the room must be instrumented, and the requirement that the cameras on the head must be able to see the ceiling (thus restricting head movement). However, these problems are not insurmountable. The majority of the bulk of their HMD is due to the size and weight of four video cameras, which are far bigger than state-of-the-art miniature cameras. When more general visual tracking algorithms are developed, the requirement that the room be instrumented will be removed. This has the additional benefit of making the tracking system "self-contained": there need only be a single device, not a transmitter and a receiver. It is conceivable that such a system, using a more general tracking algorithm that determines its position dynamically from objects in the environment, could be used in conjunction with inertial trackers and GPS (or other wireless location tracking technologies) to provide a high-accuracy tracking system with unlimited range.

Outside-in optical tracking techniques will also be useful. In the relatively near future, it will be uncommon to find a computer display without an attached video camera. The camera could be used to track a person's position relative to that display, as well as to capture video for other purposes. This is precisely the information needed to determine the position of the user's head to generate head-tracked, 3D images on the display from that person's point of view. This tracking information could also be used in other ways, as described in Section 4.3.

Regardless of the technology, tracking system latency must be improved because it is of critical importance to all virtual environments [Adelstein et al. 1993; Azuma and Bishop 1994]. All delays, measured from when a tracked object is moved until the corresponding data structures and displays are updated, adversely affect the user's experience. This is most obviously manifested in augmented reality, where any delay results in misregistration of the graphics with the real world. Many current systems do head motion prediction using extrapolating filters to reduce



(a) (b)

Figure 9: Two images of a three-dimensional axes overlaid on the corner of an object in the world. (a) An improperly registered image. (b) A properly registered image. (Courtesy of Ronald Azuma, University of North Carolina at Chapel Hill)

the apparent overall position and/or orientation lag [Liang et al. 1991]. For example, the UNC optical HMD uses an estimating filter, in addition to accelerometers and gyroscopes, to help predict both the orientation and position of the head.

4.1.2 Displays

Most of the display requirements for ubiquitous computing devices are relatively straightforward; smaller, lighter, less power consumption, higher resolution. One interesting requirement is the need to allow for the simultaneous use of multiple pens. This has been addressed to a certain degree by the Liveboard [Elrod et al. 1992]. The Liveboard is a self-contained "wall-sized" rear-projection color display system. It allows the use of multiple pens, each of which can function several meters away from the display surface. Consequently, group interaction is more natural because each user does not have to be able to reach the display surface. Such capabilities would also be useful on pad-sized displays.

The displays for virtual environments present a different set of problems, some of which have no current solution. In particular, current HMDs are not ready for widespread use. The most obvious requirement is for HMDs to have a resolution and field of view that approaches the fidelity of the human visual system. Each of our eyes has a field of view of approximately 120˚ vertically by 150˚ horizontally, the two eyes together providing a horizontal field of view of approximately 200˚ [Kalawski 1993]. A stereo HMD with one pixel per arc minute of visual angle would require each eye's display to have a resolution of 7200 by 9000 pixels! However, human visual acuity is not constant over this range, the finest details being resolved within a field of view of only 2˚ in the foveal region, which would receive a field only 120 pixels wide at this resolution. Rendering graphics at such a high resolution over our entire visual field is not necessary, as our visual acuity falls off quite rapidly as objects move out of the range of foveal vision.

A display that allows a wide field of view at low resolution and allows for the simultaneous display of a high-resolution inset that follows the user's gaze would be equally effective. This approach is used by CAE Electronics in their fiber-optic see-through HMD, which uses fiber-optic bundles to transfer images from light valve projectors to the HMD. Each eye is presented with a low-resolution image with a field of view of 88˚ by 66˚. Through use of an eye tracker, a higher resolution image, with a field of view of 24˚ by 19˚, follows the direction of regard of the user's eye [Shenker 1987]. A display such as this presents a great challenge to the rendering software to ensure that visual artifacts are not introduced when objects move into and out of the inset region.

Clearly, HMDs must be made smaller, lighter and non-tethered. See-through displays must eventually be constructed that allow individual pixels to be transparent or opaque. A more subtle problem is depth of field. In most display technologies, all pixels are focused at the same depth, whether a single image or a stereo pair is displayed. In addition to forcing our visual system to deal with two conflicting depth cues, this causes visual clutter because everything is in focus all the time. This problem is especially noticeable in augmented reality because a virtual object could be out of focus when the real object with which it is collocated is in focus. Therefore, displays must be constructed that allow individual pixels to be focused at different depths under program control.

4.1.3 Processors

If video is to play a larger role in ubiquitous computing and virtual environments, real-time manipulation of video streams must be possible. For example, consider the Starfire video presentation produced by Sunsoft to show their vision of what a next-generation computer interface might be like [Tognazzini 1994]. At one point in the presentation, the video windows displaying two remote participants in a teleconference are rotated in perspective to give the appearance that all three participants can see each other (Figure 10).

Figure 10: By rotating the images of the two remote participants in a video conference, the three participants could all see each other in the same relative positions. (Courtesy of Bruce Tognazzini, SunSoft)

While such effects are currently possible in real time using high-end special-purpose broadcast video effects systems, they could not be accomplished with digital video streams on standard, multimedia computers. High-end graphics processors allow textures to be mapped to 3D surfaces without significant overhead in the graphics pipeline [Akeley 1993]. *Creating* the texture data structures from raw images is time-consuming, however, precluding real-time modification of the texture itself on early hardware.

4.1.4 Networks

The success of ubiquitous computing hinges on the development of high-speed wireless networks that far exceed the capacity planned for commercial systems over the next decade. Commercial systems are aiming at low-speed connections over fairly wide ranges (e.g., 64kbits/sec). The huge number of ubiquitous computing devices suggests that significantly higher bandwidth will be required, and that networks to support it should be designed to provide high volume *per unit area* [Weiser 1991]. For example, Weiser suggests that in a building with 300 people, each using 100 wireless devices, each demanding 256Kbits per second, the required aggregate bandwidth is 7.5Gbits per second. This will be much easier to achieve if the networks provide high bandwidth per unit area by using a cellular system with cells the size of an office, allowing each radio frequency to be reused hundreds of times throughout the building, rather than attempting to build one 7.5Gbit long range wireless network [Weiser 1993b]. Furthermore, since many of the devices may be displaying video rather than just traditional graphics, the bandwidth requirements may be an order of magnitude higher.

Until video becomes more prevalent in virtual environments, which will happen as the display hardware begins to support it (see Section 4.3), virtual environments will have much lower bandwidth requirements, but will have much stricter latency requirements. For head-tracker data in particular, every millisecond of latency will be perceived by the user as lag in the system's response to movement.

4.2 Combining Virtual Environments and Ubiquitous Computing

While virtual environments and ubiquitous computing are often put forward as two competing paradigms, we instead believe they are complementary. The most obvious area of overlap involves desktop-based virtual reality. It is inconceivable that people will not run applications in a ubiquitous computing environment that use 3D graphics. Furthermore, assuming that tracking technology matures to a state that ubiquitous computing devices and users are tracked precisely, rather than coarsely, there is no reason to not take advantage of this information and display head-tracked 3D graphics whenever possible.

Weiser [Weiser 1991] gives an example of a non-head-mounted augmented reality display in a ubiquitous computing environment: a living room window that overlays on the user's view a representation of the paths neighbors or friends have taken that morning. While Weiser does not indicate that the user's head is being tracked, precisely registered graphics are impossible otherwise. If head-tracked graphics are presented, however, they will be generated from the viewpoint of only *one* viewer at a time, assuming the display is based on current display technologies. All other viewers would see misregistered graphics whose viewpoint would change as the tracked user moved. Fortunately, this multiple-viewer problem can be solved by using individual see-through HMDs.

There are other advantages to having much of the ubiquitous computing environment's material presented on personal eyewear. First, all devices are not required to have their own displays. Instead, they can simply notify the infrastructure that they have some virtual information that can be displayed, and each user's eyewear can present it if appropriate. This allows ubiquitous computing devices to have their displays individually customized for any number of simultaneous users. Furthermore, just as current ubiquitous computing researchers are careful about allowing people to disable personal tracking and to disable the ubiquitous computing controls in their environment [Elrod et al. 1993a; Elrod et al. 1993b], one user's eyewear can be turned off without affecting others.

The advantages of augmentation are not limited to visual displays. Having many active audio displays in an environment can be even more annoying than having many graphics displays. The use of personal spatialized audio can avoid polluting the sonic environments of other users.

We need to investigate how to accommodate large numbers of computers and their user interfaces (both physical and virtual, visual and aural) within the virtual world. Feiner and Shamash have looked at such *hybrid user interfaces*, which combine real and virtual displays in a common user interface [Feiner and Shamash 1991]. They embedded the 2D flat-panel display of a personal computer in a 3D virtual information surround presented on a see-through HMD. In their prototype hybrid user interface window manager, the flat panel served as a primary work space. Windows could be moved freely between the flat panel and the much larger, but lower-resolution, information space on the HMD, which provided context. While this work needs to be extended to account for the large number and variety of devices populating a ubiquitous computing environment, we feel the concept of hybrid user interfaces is one of the keys to the effective integration of ubiquitous computing and augmented reality. This approach provides a way of addressing the multiple-viewer problem discussed above, and supporting collaboration among collocated users in augmented reality (see Section 4.10).

4.3 The Place of Video

Video will play an increasingly important role in multimedia interfaces. One of the advantages of integrating video cameras with displays, which was highlighted in Section 4.1.1, is the potential for such displays, fixed or handheld, to track the people in front of the display. In addition to providing the necessary information for the generation of head-tracked 3D graphics, positional information can be used to determine what portion of the video image gets transmitted in video applications. It should be possible, for example, to ensure that the transmitted image contains a centered, head-and-shoulder view of the user. To achieve this, the camera must be mounted on a moving base, enabling it to always remain pointed at the user. A promising technology for this is Bederson's eyeball-like spherical pointing motor [Bederson 1992], which can accurately position the camera mounted within it at rotational speeds of several hundred degrees per second.

Using two cameras instead of one would enable the transfer of stereo images. If the cameras can be positioned appropriately for the remote viewer's interocular distance, a reasonable stereo pair of images could be displayed. Alternatively, it may be possible to construct dynamically a 3D model of the visible part of the user's head and transmit that, allowing the recipient's system to display the remote user as desired [Ohya et al. 1993].

A variety of video-based approaches are being developed to capture depth maps of the environment (e.g., [Nayar et al. 1995]). This could make it possible to mix in synthesized graphics with the view of an otherwise unfamiliar environment and have visibility relationships be properly maintained.

Such technologies notwithstanding, if traditional video is to be integrated more fully into ubiquitous computing and virtual environments, we need the ability to map real-time video streams onto 3D surfaces, as discussed in Section 4.1.3. Doing so will facilitate placing video streams in a virtual environment on arbitrarily positioned "screens."

4.4 Nonphotorealistic Rendering

As mentioned in Section 3.3, the majority of current systems strive for photorealistic rendering in their virtual worlds. However, this is not always the most appropriate approach for communicating information.

Consider technical illustrations found in most manuals, or architectural illustrations of structures. These are typically line drawings that use cut-away views and varying line-styles to convey a clearer understanding of the object or building than would a realistic rendering of the same scene. Kamada and Kawai [Kamada and Kawai 1987] discuss automated approaches to generating images in which otherwise hidden edges are rendered with different line styles depending on what objects hide them. Such an approach to rendering is important in augmented reality to convey information with a minimum of augmenting graphics. Feiner and Seligmann develop methods for automatically generating cut-away views [Feiner and Seligmann 1992]. Winkenbach and Salesin [Winkenbach and Salesin 1994] and Strothotte et al. [Strothotte et al. 1994] describe systems that produce images inspired by architectural renderings.

For immersive simulations, such as virtual walkthroughs, realism is important. However, as scenes become more complex, a significant amount of the rendering time is devoted to objects that are not perceptible to the user, either because they are hidden or because the user is not looking at them. Graphics systems have long had a variety of facilities for building graphical objects that could adapt to their transformed position and size. For example, objects can be culled from a scene or rendered at a lower level of detail, based on the size of their projections. The rendering requirements of virtual environments have rekindled the need for these facilities, and some researchers have suggested the use of culling and image degradation techniques that are based on the physiology of human perception [Gossweiler and Pausch 1994].

4.5 Integration with Existing User Interface Paradigms

Much of the user-interface research that has been done on traditional displays will remain relevant to ubiquitous computing and virtual environments since many kinds of information are most effectively displayed using 2D techniques. Furthermore, the displays that will surround us in a ubiquitous computing environment will be similar in many ways to the kinds of displays we currently use. Here we discuss how ubiquitous computing and virtual environments can benefit by incorporating some of the major 2D paradigms. We first examine *direct manipulation*, the most common technique currently in use in virtual environments. Then we shift our attention to more general issues surrounding so-called WIMP (Windows, Icons, Menus, Pointing) interfaces, as they are currently the most studied class of 2D interfaces. Finally, we discuss how 2D window-management techniques can be adapted to accomplish 3D *environment management*.

4.5.1 Direct Manipulation

Direct manipulation refers to interaction techniques in which the objects of interest are visible to the user and are manipulated directly by rapid, reversible, incremental actions instead of a complex command language [Shneiderman 1983]. Direct manipulation in 3D is often touted as a feature of virtual environments because of its supposed naturalness and ease of learning. However, reaching and grabbing in 3D can be far less convenient and effective than the corresponding 2D direct manipulation techniques, and these, in turn are often criticized for being more tedious than the command line interfaces that they replaced. The problems are compounded by the poor quality of current trackers and gloves and the almost complete lack of haptic feedback in most virtual environments.

Even with perfect hand tracking, the user must be able to grab the object to manipulate it. This implies that the object must be within arm's length. Objects that a user cannot easily grab for other reasons, such as fast moving or small objects, present similar problems. Direct manipulation in 3D can also require that the user hold their hands up without any physical support for extended periods of time. This has long been known to be a problem, dating back to the first use of touch-screens and light pens, and is further complicated by the fact that there is not even a screen against which to rest the hand.

An often-implemented solution to this problem is to give the user a "virtual flashlight" that allows them to select any object they can see and point at. However, this causes problems of its own. For example, the farther away an object is, the harder it is to select with a virtual flashlight because the user is changing angular, not Euclidean, distance when manipulating the pointer. Because of these shortcomings, we need to explore techniques that work well on objects that are distant, extremely large or small, quick moving, or otherwise difficult to manipulate directly.

4.5.2 A WIMP No More

Existing WIMP techniques and applications will have to be adapted to fit the newer devices, ranging from handhelds to head-mounts. Furthermore, the manner in which we use even the more familiar devices requires that we modify some of our user-interface techniques.

Direct use of WIMP techniques in opaque, immersive virtual environments has been attempted but is typically less successful than in 2D. Only the simplest of interaction devices can be used because the user cannot see the physical devices in the virtual world. For example, a virtual equivalent to the mouse is currently practical. In contrast, a virtual typewriter keyboard is not, because of poor tracking and lack of haptic feedback. Button pads have been successfully emulated by positioning a set of large virtual buttons in the plane of a physical desk to provide first-order haptic feedback [Weimer and Ganapathy 1989]. Other passive physical props that are instrumented with trackers can be used to provide the user with more facile control over virtual objects [Hinckley et al. 1994].

An example of adaptation of WIMP techniques is the virtual panel system provided in the MR Toolkit [Shaw et al. 1993]. An MR panel is essentially an arbitrarily oriented rectangle that can contain hierarchical menus, sliders, buttons, and text. Manipulation is accomplished based on the intersection between a ray cast from the user's hand and the surface of the panel. However, due to the problems with current trackers and gloves, compounded by lack of haptic feedback, users often experience problems using panels. More appropriate analogs of existing techniques must be found for use in immersive virtual worlds.

Non-immersive virtual environments have the advantage that the user can use traditional input devices, such as keyboards and mice. Monocular see-through systems can use the pointer devices in a straightforward fashion, by treating the view plane as a large 2D display on which the mouse cursor moves. The mouse may be used to manipulate a 3D scene by allowing objects to be manipulated anywhere along the line of sight from the eyepoint through the 2D cursor [Feiner et al. 1993a]. Unfortunately, using these devices in stereo systems is not a straightforward task. The pointer now has a three dimensional presence, so traditional techniques will not work in a straightforward manner: there are two lines of sight passing through the pointer, one from each of the user's eyes. Systems must either use 3D input devices, or use less straightforward approaches with 2D devices. For example, the 2D cursor could still move around on the 2D viewing plane and be linked to a 3D "ray," perhaps emanating from a fixed point between the user's eyes and intersecting the cursor on the viewplane.

In a ubiquitous computing environment, on the other hand, we may be using pens, voice and touchscreens more than mice and keyboards. A large body of research exists on pen-based input techniques [Goldberg and Goodisman 1991]. Tablet computers running pen-based operating systems, such as Microsoft Windows for Pen Computing or PenPoint [Carr and Shafer 1991], are becoming more common. PDAs such as the Newton have handwriting recognition that is promising, though not yet generally useful.

A major complaint about pen-based computing is that printing and writing are much slower forms of input than keyboards. Researchers are investigating ways of increasing the speed of handwritten input. Goldberg and Richardson developed a pen-based input technique using a specifically designed alphabet called *unistrokes*, that is significantly faster, and recognized input with a much higher accuracy, than handwriting or printing [Goldberg and Richardson 1993]. A alternate scheme, *T-Cube*, has each character correspond to a flick gesture determined by their placement on a set of pie menus [Venolia and Neiberg 1994]. Commercial software, such as Graffiti [Graffiti 1994], which uses a simplified alphabet for improved printing recognition, is available for PDAs such as the Apple Newton. All of these techniques are faster and more accurate than printing with a traditional alphabet, although they require additional training.

What is less understood are the differences in input techniques required when the size of the displays and the manner in which we use them is radically different. Pier and Landay point out that interfaces must be *location-independent* [Pier and Landay 1992; Landay 1993]. By this they mean interfaces should not rely on being able to do such things as reserving portions of the display for widgets (if they are to be adapted to small displays), or on the user being able to point to specific areas of the display easily, such as the top or bottom edge (if they are to be adapted to very large displays).

Finally, when many large and small displays are available per person, the manner in which each is used will change. Current graphical user interfaces are based on the premise that we have one display that is about the size of two sheets of 8.5"×11" paper, that we must share that display with many applications, and that an application will remain on the display on which it is first created. The effects of these assumptions are far reaching. An important consequence, from the user's perspective, is that the window system must provide a *window manager* to help the user manage a much larger set of windows than can fit on the display.

Unfortunately, most existing window systems are inappropriate for use in these new environments. For example, microcomputer window systems such as Microsoft Windows [Microsoft 1992] and the Apple Macintosh Desktop [Apple 1987], require that all programs run locally. This allows them to be quite responsive, but makes them unsuitable for a distributed environment. The most popular networked window system, X [Scheifler and Gettys 1986], does not provide a sufficiently general layer of abstraction between applications and the window system to allow windows to be moved between displays of the same type, not to mention moving them between displays of different sizes and capabilities. Various attempts have been made to modify X to rectify some of these problems [Kantarjiev et al. 1993], but none have been entirely satisfactory. By and large, they allow windows to be moved between virtually identical displays; however, without modifying the applications, more transparent window mobility is difficult to achieve. Other window systems, such as NeXTSTEP [NeXTSTEP 1992], NeWS [Sun 1991] and Trestle [Manasse and Nelson 1991], provide a layer of abstraction between applications and the window system, but do not address the requirements of moving windows between significantly different displays. Developing a suitable window system while providing the capabilities of existing systems will be a significant challenge.

4.5.3 Environment Management

In the previous section, we discussed the problems encountered when attempting to use current windowing systems in a ubiquitous computing and virtual environment. One of the prime motivations for trying to use an existing window system, rather than inventing a new, more appropriate one, is the desire to support the huge number of existing and future applications [Feiner et al. 1993a]. When adapting the windowing system to these new environments, many of the issues that must be addressed involve window management. Screen real estate will not be at a premium, as it is currently. Rather, there will be a surplus of real estate spread across hundreds of devices. Users will not need help organizing 2D windows on a small crowded 2D display; instead, they will need help in organizing (and finding) orders of magnitude more 2D and 3D objects scattered about the environment. Thus we need to extend the notion of window management to that of *environment management.*

A simple problem with current window managers is that, for the most part, they force users to position all windows manually. Automatic positioning of windows is nontrivial because of the overcrowded displays. In ubiquitous computing and augmented reality, we potentially have far more display real estate, allowing environment managers much more freedom to make reasonable choices about object placement. In augmented reality, however, there is the potential for an overlaid object to block something that is also of interest, such as a display or another person [Feiner et al. 1993a]. The environment manager should help users avoid these problems by moving existing virtual objects when necessary.

Different approaches to window management will be needed for pen-based interfaces. For example, the "piles" metaphor explored by Mander, Salomon and Wong [Mander et al. 1992] displays a set of documents as a small, messily stacked pile of paper. This is a more meaningful representation of a set of documents than is a "folder" on a typical interface, such as the Macintosh, because it conveys not only the existence of a group of documents, but also its approximate size, contents and relative order. The piles metaphor takes advantage of our ability to form and recognize patterns in complicated data. Different document piles would soon take on meaning for their owner by virtue of their shape and size, without the need for manual labeling.

4.6 Learning from Other Disciplines

The user-interface community in general needs to pay attention to what is happening in other disciplines. This is becoming increasingly important as user interfaces become more "intelligent" and attempt to do more for the user.

There is much we can learn from other disciplines that are older and more mature than our own. With continued increases in compute power and improvements in interaction devices, we can more readily take advantage of the experience of others. For example, we can learn from animators (who use motion blur, squash and stretch, and other techniques [Chang and Ungar 1993]), perceptual and cognitive psychologists (who study how people understand sensory phenomena), and architects and landscape designers (who structure real 3D spaces for effective use by their occupants).

4.7 Integration with Hypermedia

Meyrowitz [Haan et al. 1992] stresses the importance of making hypertext capabilities available throughout the operating system, rather than in separate hypertext applications. We need to extend this notion of "universal hypertext" linking to make it possible to link arbitrary physical and virtual objects. Feiner, MacIntyre, Haupt, and Solomon [Feiner et al. 1993a] have developed a prototype augmented reality system that allows hypertextual links to be created between X windows and physical objects in the environment. For example, this could make it possible to automatically find notes you made on various books on your shelf, attach reminders to people and locations similar to the way calendar programs attach them to dates and times, or attach hypertextual links to yourself that others could follow when you are in the room.

4.8 Authoring

While authoring material in any individual medium is both difficult and time-consuming in itself, creating a quality multimedia document requires expertise in all the individual media used, as well as skill in making these media work together effectively. In fact, the effort that it takes to author a multimedia document is rarely mentioned, and when it is, is sometimes attributed solely to deficiencies in the editor(s) being used, rather than the process itself.

One research direction that addresses these difficulties is the automated design of multimedia, a goal explored in the COMET [Feiner and McKeown 1991] and WIP [Wahlster et al. 1991] knowledge-based multimedia generation projects. These systems use AI techniques to create explanatory text and graphics that fulfill a set of input goals, and coordinate the material generated in different media (e.g., by making it possible for generated text to refer to generated graphics [McKeown et al. 1992]).

Authoring is made yet harder still when the media include the full set of possibilities available in a dynamic virtual environment. *AutoVisual* [Beshers and Feiner 1993] and KARMA [Feiner et al. 1993b] address different aspects of creating virtual environments automatically. *AutoVisual* designs desktop virtual environments that present visualizations of multivariate data, taking into account criteria such as the need for interactive responsiveness. For example, it controls the amount of time that a display takes to render by varying the resolution at which representations of data are calculated and displayed. One component of KARMA guides its users through an equipment maintenance task by creating graphics that overlay the user's view of the physical world. KARMA's design can take into account the position and orientation of both the user's head and the objects being documented, so the information it presents can be redesigned as the user moves. For example, an annotation may appear only if the object being annotated is physically obscured by another object. It will be a major challenge to develop robust virtual environment design systems that can handle an even broader range of media in a production environment, including temporal media, such as animation and speech.

4.8.1 Agents

In situations where the design of a multimedia presentation should not be completely automated, "agents" have been proposed as a useful framework for the development of tools to assist with the creation and presentation of information [Riecken 1994]. For example, agents can assist with the collection of information by monitoring electronic newswires for "interesting" articles or roaming computer networks to gather information from remote databases. They could even be charged with carrying out tasks on the part of their owner. Other agents could provide assistance during video editing by processing unannotated footage to recognize cuts, pans, zooms, or find all shots including some participant or location [Ueda et al. 1991].

As suggested in the previous section, the full set of output possibilities available in dynamic virtual and ubiquitous computing environments greatly complicates a multimedia presentation. Merely selecting an output device from among the many available displays (with possibly widely varying characteristics) is a non-trivial task. By using the environmental knowledge available in a ubiquitous computing environment, display agents could automatically select displays for output, or migrate information between displays. Agents can also assist with organizing information on a single display, such as an augmented reality overlay or a large wall-sized display.

4.9 Capturing and Journaling Personal Experience

In 1945, Vannevar Bush [Bush 1945] suggested that scientists would someday wear small head-mounted cameras and microphones that would allow them to selectively photograph things of interest they were inspecting, and enter notes and annotations vocally as they worked in their labs. Technology will make it possible to extend this vision to that of capturing everything that the user can see, hear, and otherwise experience. For example, a tiny pair of cameras and microphones embedded in the frames of a pair of glasses could be used. Improvements in compression techniques and decreases in memory cost and size will eventually make this idea feasible. This raises a myriad of issues about how one might index, sort, edit, and search through a lifetime of recorded experience.

4.10 Collaboration

Virtual environments and ubiquitous computing raise additional issues beyond those addressed by current research in collaboration and CSCW. In particular, we must support collaborating users with different display and interaction devices and paradigms, and mobile participants. For example, of two collaborating users, one may have an HMD, while another may have a hand-held PDA.

Ubiquitous computing devices can provide sharable, high resolution displays on which to display shared information in an augmented reality application. It will also be necessary to support private views of information in multi-user virtual worlds. For example, if shared presence means being able to see some representation of the other user, how much of the other user's virtual environment do we have to be able to see? The classical "shared pointer" metaphor will not always work if each user's view of the shared information can be different.

4.11 Privacy

The privacy issues of ubiquitous computing and virtual environments are significant. If we are tracked everywhere we go, the volume of data that could be collected is staggering. Weiser discusses the implications of wearing active badges, such as the potential for an interested person to see how long someone spends in any room in the building [Weiser 1993a]. While active badges can track the user to the granularity of a room, high-resolution trackers can potentially let us know everything that a user is doing all the time.

The possibilities of what an employer could do with such information are not pleasant. Implementors need to pay special attention to ensure that privacy can be maintained. Weiser suggests that a combination of technological and social solutions is best. For example, employers can currently use video cameras and computer activity monitoring to watch their employees if they desire, but in general the social and legal climate often discourages the use of these technically feasible measures.

Since no technical solution for ensuring privacy in a ubiquitous computing environment is likely to succeed without seriously hampering the utility of the system, current attitudes toward employee and personal attitudes must adjust appropriately. Global activity monitoring of individuals may be useful in many obvious ways, such as automatic forwarding of phone calls. More interesting applications exist as well. For example, an automatic diary could be generated that records not only what a user did and where they did it, but who was in the same area throughout the day [Newman et al. 1992]. This could make it much easier to fill out time sheets, remember who was involved in meetings, or even check when you last went to the cafeteria. Unfortunately, if you can check to see who it was you talked to on the way back from the cafeteria, so could someone with less innocent intentions unless an information security system is in place to prevent it.

Other approaches besides uniform global availability could be taken to provide this information to others. For example, you could make your positional information available to anyone who was physically near you, so someone you talked to in the hallway would later be able to look back and remember it was you. However, they would not necessarily be able to find out where you were before or after that meeting. But, this still raises the possibility that if enough people pooled their information they could reconstruct much of your life! This underscores Weiser's point that technical solutions to privacy will never be enough; the ease with which electronic information can be pooled requires that the problem must be addressed socially as well. Imagine "location information service bureaus" paying people for the contact information they collect, pooling that information and then selling it to whomever is willing to pay.

These privacy requirements raise interesting interface and database questions as well: How would someone easily specify who can know what parts of the huge volume of information collected about them over the day? How would a query such as "Whom did I meet on my way back from the cafeteria this afternoon?" be stated and answered, in light of the possibly complicated combination of public and private data?

Another privacy issue that could be addressed by technology is eavesdropping on confidential information. Current interfaces provide no way to prevent someone who can see a display from reading it. For example, there is no way for someone working in a public place to protect themselves from eavesdroppers. It is easy to read the display of a person sitting beside or in front of you on a plane or train. Augmented reality provides a further solution to this problem. By displaying a virtual image that only the wearer of the HMD can see, the possibility of someone reading over your shoulder is eliminated. In the collaborative environment described in the previous section, selected parts of the personal information that augments public displays can be kept confidential.

Ubiquitous computing can also provide straightforward solutions in other cases. For example, if a document under group discussion is confidential, such as an employee's performance evaluation, it could automatically close if that employee, or another inappropriate viewer, entered the room.

5 Conclusions

We have surveyed the state of the art in virtual environments and ubiquitous computing, two user interface paradigms that will shape the future of computing. We have tried tro show how they complement each other, and how they provide new capabilities for multimedia user interfaces, and we have suggested a set of issues for multimedia researchers to explore.

These paradigms differ from those prevalent in current multimedia systems, in part because they use new hardware technologies: 3D displays and interaction devices, in the case of virtual environments, and large numbers of computers integrated into the environment, for ubiquitous computing. Furthermore, both paradigms impose significant additional requirements on operating systems and networks than do current desktop-based multimedia applications. For example, virtual environments often involve the transmission of information generated by 3D trackers. While this is a relatively small amount of information in comparison with digital video, its latency requirements are much stricter if multiple users are to interact effectively with each other. In comparison, ubiquitous computing will require high bandwidth over extremely short distances, resulting in extremely high aggregate bandwidth over an entire network.

As a consequence of these new trends, many of the assumptions being made in multimedia support systems will have to be reexamined. These range from current user interface metaphors, such as the video-enhanced WIMP approach, to current computing infrastructure, such as relatively static network configurations.

6 References

ADELSTEIN, B. D., JOHNSTON, E. R., AND ELLIS, S. R. 1993. A Testbed for Characterizing Dynamic Response of Virtual Environment Spatial Sensors. In *Proc. ACM UIST '93*, pages 15–22.

AKELEY, K. 1993. RealityEngine Graphics. In *Computer Graphics (Proc. ACM SIGGRAPH '93)*, Annual Conference Series, pages 109–116.

APPLE 1987. *Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley Publishing Co., Reading, MA.

AZUMA, R. 1993. Tracking Requirements for Augmented Reality. *CACM*, 36(7):50–51.

AZUMA, R. AND BISHOP, G. 1994. Improving Static and Dynamic Registration in an Optical See-through HMD. In *Computer Graphics (Proc. ACM SIGGRAPH '94)*, Annual Conference Series, pages 197–204.

BAJURA, M., FUCHS, H., AND OHBUCHI, R. 1992. Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery Within the Patient. In *Computer Graphics (Proc. ACM SIGGRAPH '92)*, Annual Conference Series, pages 203–210.

BEDERSON, B. 1992. *A Miniature Space-Variant Active Vision System: Cortex-I*. PhD dissertation, New York University, New York, NY. Also available as Technical Report No. 611.

BEGAULT, D. 1994. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, Boston, MA.

BESHERS, C. AND FEINER, S. 1993. AutoVisual: Rule-Based Design of Interactive Multivariate Visualizations. *IEEE Computer Graphics and Applications*, 13(4):41–49.

BROOKS, JR., F. 1988. Grasping Reality Through Illusion—Interactive Graphics Serving Science. In *Proc. ACM CHI '88*, pages 1–10.

BROOKS, JR., F. P., OUH-YOUNG, M., BATTER, J. J., AND KILPATRICK, P. J. 1990. Project GROPE—Haptic Displays for Scientific Visualization. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 177–185.

BUSH, V. 1945. As We May Think. *The Atlantic Monthly*, 176(1):101–108.

CALVIN, J., DICKENS, A., GAINES, B., METZGER, P., MILLER, D., AND OWEN, D. 1993. The SIMNET Virtual World Architecture. In *Proc. IEEE VRAIS '93*, pages 450–455.

CARLSSON, C. AND HAGSAND, O. 1993. DIVE—a Multi-User Virtual Reality System. In *Proc. IEEE VRAIS '93*, pages 394–400.

CARR, R. AND SHAFER, D. 1991. *The Power of Penpoint*. Addison-Wesley, New York, NY.

CHANG, B.-W. AND UNGAR, D. 1993. Animation: From Cartoons to the User Interface. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, Annual Conference Series, pages 45–55.

CODELLA, C. F., JALILI, R., KOVED, L., AND LEWIS, J. B. 1993. A Toolkit for Developing Multi-User, Distributed Virtual Environments. In *Proc. IEEE VRAIS '93*, pages 401–407.

CRUZ-NEIRA, C., SANDIN, D. J., AND DEFANTI, T. A. 1993. Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE. In *Computer Graphics (Proc. ACM SIGGRAPH '93)*, Annual Conference Series, pages 135–142.

DEERING, M. F. 1993. Making Virtual Reality More Real: Experience with the Virtual Portal. In *Proc. Graphics Interface '93*, pages 219–226.

DUBBERLY, H. AND MITCH, D. 1987. The Knowledge Navigator. Video. Apple Computer, Inc.

ELROD, S., BRUCE, R., GOLD, R., GOLDBERG, D., HALASZ, F., JANSSEN, W., LEE, D., MCCALL, K., PEDERSEN, E., PIER, K., TANG, J., AND WELCH, B. 1992. Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration. In *Proc. ACM CHI '92*, pages 599–607. Also available as Xerox PARC Tech Report CSL-92-6.

ELROD, S., HALL, G., COSTANZA, R., DIXON, M., AND DES RIVIERES, J. 1993a. Responsive Office Environments. *CACM*, 36(7):84–85.

ELROD, S., HALL, G., COSTANZA, R., DIXON, M., AND DESRIVIERES, J. 1993b. The Responsive Environment: Using Ubiquitous Computing for Office Comfort and Energy Management. Technical Report CSL-93-5, Xerox Palo Alto Research Center.

FEINER, S., MACINTYRE, B., HAUPT, M., AND SOLOMON, E. 1993a. Windows on the World: 2D Windows for 3D Augmented Reality. In *Proc. ACM UIST '93*, pages 145–155.

FEINER, S., MACINTYRE, B., AND SELIGMANN, D. 1993b. Knowledge-Based Augmented Reality. *CACM*, 36(7):52–63.

FEINER, S. AND MCKEOWN, K. 1991. Automating the Generation of Coordinated Multimedia Explanations. *IEEE Computer*, 24(10):33–41.

FEINER, S. AND SELIGMANN, D. 1992. Cutaways and Ghosting: Satisfying Visibility Constraints in Dynamic 3D Illustrations. *The Visual Computer*, 8(5–6):292–302.

FEINER, S. AND SHAMASH, A. 1991. Hybrid User Interfaces: Breeding Virtually Bigger Interfaces for Physically Smaller Computers. In *Proc. ACM (UIST) '91*, pages 9–17.

FISHER, S., MCGREEVY, M., HUMPHRIES, J., AND ROBINETT, W. 1986. Virtual Environment Display System. In *Proc. 1986 ACM Workshop on Interactive 3D Graphics*, pages 77–87.

FITZMAURICE, G. W. 1993. Situated Information Spaces and Spatially Aware Palmtop Computers. *CACM*, 36(7):38–49.

FUCHS, H., LEVOY, M., AND PIZER, S. M. 1990. Interactive visualization of 3D medical data. *Visualization in Scientific Computing*, pages 140–146. Reprinted from IEEE Computer, August 1989, pp. 46–51.

FUKUMOTO, M. AND SUENAGA, Y. 1994. FingeRing: A Full-Time Wearable Interface. In *Proc. ACM CHI '92*, pages 81–82.

GAERTNER 1992. The GRD-1000 Series Position Tracking System. Promotional literature from Gaertner Research Division, GEC Ferranti Defense Systems Inc.

GETTING, I. A. 1993. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47.

GOLDBERG, D. AND GOODISMAN, A. 1991. Stylus User Interfaces For Manipulating Text. In *Proc. ACM UIST '91*, pages 127–135.

GOLDBERG, D. AND RICHARDSON, C. 1993. Touch-Typing with a Stylus. In *Proc. ACM INTERCHI '93*, pages 80–87.

GOLDBERG, D. AND TSO, M. 1993. How to Program Networked Portable Computers. In *Proc. Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 30–33.

GOSSWEILER, R. AND PAUSCH, R. 1994. A System for Application-Independent Time-Critical Rendering. In *Proc. ACM CHI '94*, pages 261–262.

GRAFFITI 1994. Graffiti Text Input Software for the Apple Newton. Palm Computing, Los Altos, CA.

GRIMSDALE, G. 1991. dVS—Distributed Virtual Environment System. In *Proc. Computer Graphics '91 Conference*.

HAAN, B. J., KAHN, P., RILEY, V. A., COOMBS, J. H., AND MEYROWITZ, N. K. 1992. IRIS Hypermedia Services. *CACM*, 35(1):36–51.

HEILIG, M. L. 1960. Stereoscopic-Television Apparatus for Individual Use. United States Patent Number 2,955,156. Filed May 24, 1957. Also Reprinted in Computer Graphics, 28 (2), May 1994, pages 131–134.

HEILIG, M. L. 1962. Sensorama Simulator. United States Patent Number 3,050,870. Filed August 28, 1962.

HINCKLEY, K., PAUSCH, R., GOBLE, J. C., AND KASSELL, N. F. 1994. Passive Real-World Interface Props for Neurosurgical Visualization. In *Proc. ACM CHI '94*, pages 452–458.

IOANNIDIS, J. AND MAGUIRE, JR., G. Q. 1993. The Design and Implementation of a Mobile Internetworking Architecture. In *Proc. 1993 Winter USENIX*, pages 491–502.

KALAWSKI, R. S. 1993. *The Science of Virtual Reality and Virtual Environments*. Addison-Wesley, Reading, MA.

KAMADA, T. AND KAWAI, S. 1987. An Enhanced Treatment of Hidden Lines. *ACM Transactions on Graphics*, 6:308–323.

KANTARJIEV, C. A., DEMERS, A., FREDERICK, R., KRIVACIC, R. T., AND WEISER, M. 1993. Experiences with X in a Wireless Environment. In *Proc. USENIX Symposium on Mobile & Location-Independent Computing*, pages 117–128.

KATZ, W. 1994. Military Networking Technology Applied to Location-Based, Theme Park and Home Entertainment Systems. *Computer Graphics*, 28(2):110–112.

KAY, A. 1977. Microelectronics and the Personal Computer. *Scientific American*, 237(3):231–244.

KAZMAN, R. 1993. Making WAVES: On the design of architectures for low-end distributed virtual environments. In *Proc. IEEE VRAIS '93*, pages 443–449.

KRUEGER, M. W. 1991. *Artificial Reality II*. Addison-Wesley Publishing Co., Reading, MA.

KRUEGER, M. W. 1993. Environmental Technology: Making the Real World Virtual. *CACM*, 36(7):36–37.

LANDAY, J. 1993. User Interface Issues in Mobile Computing. In *Proc. Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 40–47.

LIANG, J., SHAW, C., AND GREEN, M. 1991. On Temporal-Spatial Realism in the Virtual Reality Environment. In *Proc. ACM UIST '91*, pages 19–25.

LIPPMAN, A. 1980. Movie-Maps: an Application of the Optical Videodisc to Computer Graphics. *Computer Graphics*, 14(3):32–42.

MANASSE, M. S. AND NELSON, G. 1991. Trestle Reference Manual. Technical report, Digital Systems Research

Centre. Research Report 68.

MANDER, R., SALOMON, G., AND WONG, Y. Y. 1992. A 'Pile' Metaphor for Supporting Casual Organization of Information. In *Proc. ACM CHI '92*, pages 627–634.

MCKEOWN, K., FEINER, S., ROBIN, J., SELIGMANN, D., AND TANENBLATT, M. 1992. Generating Cross-References for Multimedia Explanation. In *Proc. AAAI '92*, pages 9–16.

MICROSOFT 1992. *The Windows Interface: An Application Design Guide*. Microsoft Press, Redmond, WA.

MINSKY, M., OUH-YOUNG, M., STEELE, O., BROOKS, JR., F. P., AND BEHENSKY, M. 1990. Feeling and Seeing: Issues in Force Display. In *Proc. 1990 ACM Symp. on Interactive 3D Graphics*, pages 235–243.

NAYAR, S. K., WATANABE, M., AND NOGUCHI, M. 1995. Real-Time Focus Range Sensor. *Proc. of Intl. Conf. on Computer Vision*, pages 995–1001.

NEWMAN, W., ELDRIDGE, M., AND LAMMING, M. 1992. PEPYS: Generating autobiographies by automatic tracking. Technical report, Xerox EuroPARC.

NEXTSTEP 1992. *NeXTSTEP User Interface Guidelines*. Addison-Wesley Publishing. Release 3.

NOLL, A. M. 1972. Man-Machine Tactile Communication. *SID Journal*, 1(2):5–11, and 30.

OHYA, J., KITAMURA, Y., TAKEMURA, H., KISHINO, F., AND TERASHIMA, N. 1993. Real-time Reproduction of 3D Human Images in Virtual Space Teleconferencing. In *Proc. IEEE VRAIS '93*, pages 408–414.

PIER, K. AND LANDAY, J. 1992. Issues for Location-Independent Interfaces. Technical Report ISTL-92-4, Xerox Palo Alto Research Center.

REGAN, M. AND POSE, R. 1993. An Interactive Graphics Display Architecture. In *Proc. IEEE VRAIS '93*, pages 293–299.

RIECKEN, D. 1994. Intelligent Agents. *CACM*, 37(7):18–21.

SCHACHTER, B. J. 1983. *Computer Image Generation*. John Wiley, New York.

SCHEIFLER, R. W. AND GETTYS, J. 1986. The X Window System. *ACM Transactions on Graphics*, 5(2):79–109.

SCHILIT, B. N., ADAMS, N., GOLD, R., TSO, M., AND WANT, R. 1993a. The sc ParcTab Mobile Computing System. In *Proc. Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 34–39.

SCHILIT, B. N., THEIMER, M. M., AND WELCH, B. B. 1993b. Customizing Mobile Application. In *Proc. USENIX Symposium on Mobile & Location-Independent Computing*, pages 129–138.

SHAW, C., GREEN, M., LIANG, J., AND SUN, Y. 1993. Decoupled Simulation in Virtual Reality with the MR Toolkit. *ACM Transactions on Information Systems*, 11(3):287–317.

SHENKER, M. 1987. Optical design criteria for binocular helment mounted displays. In *Display System Optics, SPIE Proceedings*, volume 778.

SHNEIDERMAN, B. 1983. Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16(8):57–69.

STROTHOTTE, T., PREIM, B., RAAB, A., SCHUMANN, J., AND FORSEY, D. 1994. How to render frames and influence people. *Computer Graphics Forum*, 13(3):455–466.

SUN 1991. *The NeWS Toolkit Reference Manual*. Sun Microsystems, Inc. Part Number: 800-5543-10.

SUTHERLAND, I. 1965. The Ultimate Display. In *Proc. IFIP 1965*, pages 506–508.

TAYLOR, II, R. M., ROBINETT, W., CHI, V. L., BROOKS, JR., F. P., WRIGHT, W. V., WILLIAMS, R. S., AND SNYDER, E. J. 1993. The Nanomanipulator: A Virtual Reality Interface for a Scanning Tunnelling Microscope. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 127–134.

THEIMER, M., DEMERS, A., AND WELCH, B. 1993. Operating System Issues for PDAs. In *Proc. Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 2–8.

TOGNAZZINI, B. 1994. The "Starfire" Video Prototype Project: A Case History. In *Proc. ACM CHI '94*, page 206.

TSO, M. 1993. Using Property Specifications to Achieve Graceful Disconnected Operation in an Intermittent Mobile Computing Environment. Technical Report CSL-93-8, Xerox Palo Alto Research Center.

UEDA, H., MIYATAKE, T., AND YOSHIZAWA, S. 1991. IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System. In *Proc. ACM CHI '91*, pages 343–350.

VENOLIA, D. AND NEIBERG, F. 1994. T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. In *Proc. ACM CHI '94*, pages 265–270.

VINCENT, V. AND WYSHYNSKI, S. 1993. Full-body Unencumbered Immersion in Virtual Worlds. In Wexelblat, A., editor, *Virtual Reality: Applications and Explorations*. Academic Press.

WAHLSTER, W., ANDRE, E., GRAF, W., AND RIST, T. 1991. Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation. In *Proc. European Chapter of the Assoc. for Computational Linguistics*, pages 8–14.

WANT, R., HOPPER, A., FALCAO, V., AND GIBBONS, J. 1992. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102.

WARP 1994. *VTV Developer's Kit Reference Manual*. Warp California, Inc., Sausalito, CA.

WEIMER, D. AND GANAPATHY, S. 1989. A Synthetic Visual Environment with Hand Gesturing and Voice Input. In *Proc. ACM CHI '89*, pages 235–240.

WEISER, M. 1991. The Computer for the 21st Century. *Scientific American*, 265(3):94–104.

WEISER, M. 1993a. Some Computer Science Issues in Ubiquitous Computing. *CACM*, 36(7):74–83.

WEISER, M. 1993b. Ubiquitous Computing. *IEEE Computer*, 26(10):71–72.

WELLNER, P. 1993. Interacting with paper on the DigitalDesk. *CACM*, 36(7):86–95.

WELLNER, P., MACKAY, W., AND GOLD, R. 1993. Introduction. *CACM*, 36(7):24–26.

WENZEL, E. AND FOSTER, S. 1990. Realtime Digital Synthesis of Virtual Acoustic Environments. In *Proc. 1990 ACM Symp. on Interactive 3D Graphics*, pages 139–140.

WINKENBACH, G. AND SALESIN, D. 1994. Computer-Generated Pen-and-Ink Illustration. In *Computer Graphics (Proc. ACM SIGGRAPH '94)*, Annual Conference Series, pages 91–100.

ZIMMERMAN, T., LANIER, J., BLANCHARD, C., BRYSON, S., AND HARVILL, Y. 1987. A Hand Gesture Interface Device. In *Proc. ACM CHI + GI '87*, pages 189–192.

ZYDA, M. J., PRATT, D. R., MONAHAN, J. G., AND WILSON, K. P. 1992. NPSNET: Constructing a 3D Virtual World. In *Proc. 1992 ACM Symp. on Interactive 3D Graphics*, pages 147–156.