3D User Interfaces: New Directions and Perspectives

Doug A. Bowman ▪ *Virginia Tech*

Sabine Coquillart ▪ *INRIA Grenoble Rhône-Alpes—LIG*

Bernd Froehlich ▪ *Bauhaus-Universität Weimar*

Michitaka Hirose ▪ *University of Tokyo*

Yoshifumi Kitamura and Kiyoshi Kiyokawa ▪ *Osaka University*

Wolfgang Stuerzlinger ▪ *York University*

Three-dimensional user interfaces (3D UIs) let users interact with virtual objects, environments, or information using direct 3D input in the physical and/or virtual space. In this article, the founders and organizers of the IEEE Symposium on 3D User Interfaces reflect on the state of the art in several key aspects of 3D UIs and speculate on future research.

Unless you've been living underground for the last couple of years, you know that the Nintendo Wii has taken the gaming world by storm. Wii consoles, games, and accessories fly off the shelves faster than they can be restocked, and enterprising resellers make a tidy profit hawking the Wii on eBay. Not only that, the Wii has brought a new demographic to gaming. Its appeal isn't limited to males ages 15 to 30; moms, older adults, and whole families also enjoy the games. The Wii's unique style of input and the types of games that can use this input make gaming on the Wii a unique experience.

What makes the Wii special is its 3D user interface (3D UI). It employs not only 3D graphics (like all modern gaming consoles) but also innovative spatial-input devices that can sense how the user moves them. The gamer can swing his or her arm to roll a bowling ball, point directly at the screen to grab an object, or punch the air to win a boxing match.

Although playing with the Wii is the first time that many people have seen or experienced a 3D UI, research in this area has been around for many years. Researchers in fields such as VR and augmented reality (AR), human-computer interaction, computer graphics, and human-factors engineering have all wrestled with difficult questions about the design, evaluation, and application of 3D UIs.

What 3D interaction techniques work best for important tasks such as navigation and manipulation? How should we design 3D input devices? What are the most appropriate mappings between 3D input devices, displays, and interaction techniques? How can we integrate multiple 3D techniques into a seamless 3D UI? These questions, and many others, make 3D UIs an exciting area with a wide variety of open issues.

As the Wii demonstrates, 3D UI research is more relevant than ever. As a result, the 3D UI community has been expanding and coalescing—the IEEE Symposium on 3D User Interfaces, first held in 2006, is one piece of evidence for this. In this article, leading experts in the field (the founders and organizers of the 3DUI Symposium) present seven pieces on the state of the art of 3D UIs and their future prospects.

The first four pieces describe some of the latest 3D UI research trends. Bernd Froehlich covers the design of 3D input devices, specifically the use of novel combinations of sensors. Michitaka Hirose describes using biosignals (for example, brain activity) as an input mechanism. Providing haptic

 Published by the IEEE Computer Society

(touch) feedback in 3D UIs has been a difficult topic; Sabine Coquillart describes research on pseudo-haptic interfaces, in which clever use of other sensory displays simulates haptic feedback. Yoshifumi Kitamura explores the application of 3D UIs to multidisplay interfaces.

The next two pieces provide new perspectives on 3D UI design. Wolfgang Stuerzlinger proposes eight guidelines for designing next-generation 3D interaction techniques. Another challenge is the design of multiuser, collaborative 3D UIs; Kiyoshi Kiyokawa presents several strategies addressing this challenge.

Finally, Doug Bowman provides a new perspective on 3D UI research directions. He suggests two broad strategies for increasing this research's impact.

Multisensory Input for Improved Control and Performance

Today, 3D interaction in games, CAD, or 3D animation applications is performed mainly with the 2D mouse. Everything is mapped to the input from a single pointer, so users must learn the transformations of their input actions. The more complex these transformations are, however, the harder the training and performing can be. The current trend toward multitouch interfaces at least acknowledges that humans tend to act with more than one finger at a time. However, this still only scratches the surface of the immersive experience that virtual environments will offer in future computer applications. What about grasping, turning, pushing, throwing, and jumping when interacting with applications? The Wii's success shows that users want a more engaging computer experience. In particular, professional applications are still far from providing sufficiently versatile user interfaces.

Task-Driven Design of Interaction Techniques

For control and efficiency, the user's focus must be kept on the task at hand. Graphical widgets, however, require users to keep track of their current interaction state. This is cumbersome and preempts cognitive capacities that would otherwise be available for solving the task. To improve the situation, we need to design not only software interfaces but also sensor hardware that fits spatial interaction's specific requirements. The visual paradigm "what you see is what you get" (WYSIWYG) should become an action-based approach following the idea, "what you do is what happens."

Designing human-computer interfaces in this way requires knowledge of various disciplines, including psychology, software engineering, and product design. The challenge is to find the best solution for a certain task instead of developing a workaround to enable the desired functionality in a given infrastructure. Our research at the Bauhaus-Universität Weimar proceeds in six main steps:

1. Observe the cognitive, perceptual, and motor performance of humans interacting in the physical world.
2. Model the cognitive, perceptual, and motor demands of a certain task to create interaction metaphors.
3. Develop sensors, low-level interfaces, and device drivers to record human actions as input for computer applications.
4. Design input devices—a combination of sensors assembled ergonomically.
5. Implement the designed interaction systems in prototype applications to involve users in the development process.
6. Examine usability and adjust the design.

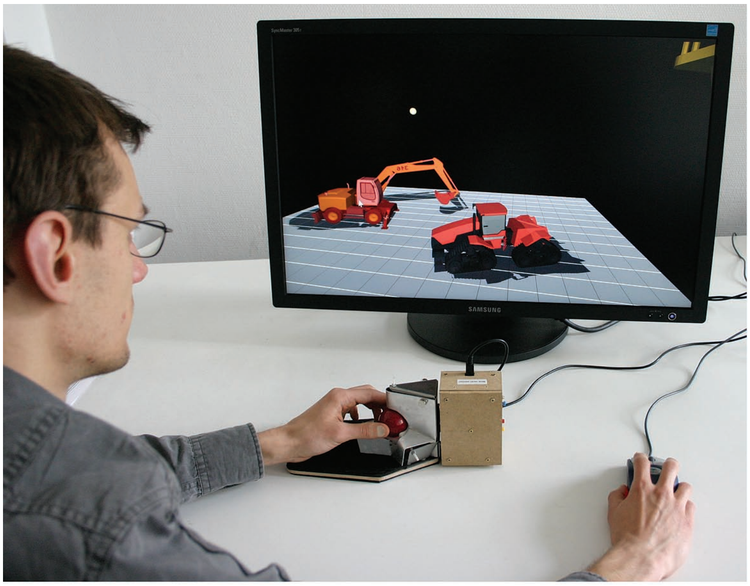
Because these aspects are interrelated, the whole design process is iterative.

Input Device Design

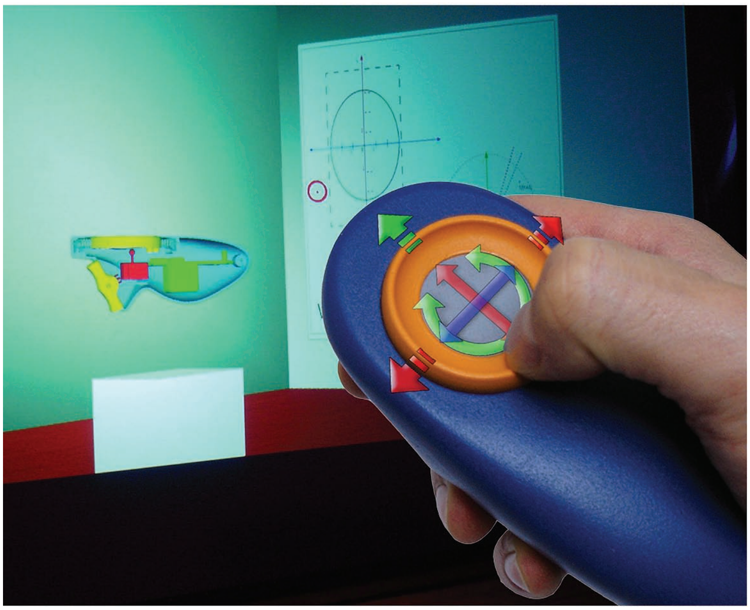
Each type of input sensor provides specific sensory feedback, depending on its construction and measured input parameters. For example, an elastic joystick's counterforce perfectly matches the task of controlling velocity during navigation through a virtual environment. Props such as Ken Hinckley's doll's head[1] resemble a manipulated virtual object and are typically free-moving devices ideally suited for positioning the object in space. Such isotonic input relies mainly on proprioception and tactile feedback. Elastic input and isometric input employ the human's force sensors (for example, tendons and ligaments). Most input devices combine different types of input sensors. We use the term *multisensory input* to emphasize that each input sensor possibly relies on different human senses or sensors.

Another major factor in the design of interaction systems is the simultaneously available degrees of freedom (DOF) of the input controller versus the integral attributes of the task it's designed to control.[2] For example, if a task requires movement in all three dimensions, the input device should support these translations along multiple axes simultaneously. If the task requires only two dimensions, as with viewpoint orientation in space, the input device's operational axes should be constrained to prevent unintentional actions.

Obviously, designing an individual controller for every type of 3D task would be uneconomical. To design input devices, we examine various tasks



(a)



(b)

Figure 1. Two multisensor 3D input devices developed at Bauhaus-Universität Weimar: (a) With the GlobeFish, the user's left hand navigates and manipulates objects with 6 degrees of freedom (DOF), and the right hand performs pointing and selection. (b) The Two-4-Six consists of a Groovepad, operated by the thumb, and an analog rocker joystick operated by the pointer finger. The Groovepad controls 5-DOF navigation; the joystick controls backward and forward motion.

in targeted applications and identify task-specific interactions and interaction sequences. These observations form the basis for our design decisions. A task's parameters define the types of sensor best employed for manufacturing a relevant control device. The relative frequency of specific tasks and the transitions between tasks provide information about which sensors to incorporate in a device and how easy switching among those sensors should be. Providing good combinations of simultaneously and separately available DOF through an ergonomic arrangement of various sensors still remains a considerable challenge. In addition—as for any physical tool—design qualities such as weight, shape, and appearance qualify input devices for certain uses. The GlobeFish and Groovepad illustrate our attempts to address these design concerns.

The GlobeFish

Manipulating objects in 3D is a central task in most digital-content-creation systems. We observed users performing this task while using an integrated 6-DOF input device, the commercially available SpaceMouse. They alternated between rotating and translating and rarely used both operations simultaneously. So, we decided to build an input device that uses separate sensors for these two interaction modes and allows rapid switching between them.

This is the central idea of our GlobeFish, a custom 3-DOF trackball embedded in a spring-loaded frame (see Figure 1a).[3] The trackball sensor measures the rotation of the ball, which is manipulated by the fingertips, and transforms the sensor reading into a corresponding rotation of a virtual object. Tightening the grip on the trackball and pushing or pulling it in any direction controls the virtual object's translation along the corresponding spatial dimensions. In a user study, we compared GlobeFish to the SpaceMouse for object positioning.[3] The GlobeFish clearly outperformed the SpaceMouse, and most users preferred the new device.

Motivated by these results, we're studying GlobeFish's usability for viewpoint navigation. Because this task is more complex than manipulation performance, it can't be evaluated as easily. Navigation in large environments involves motor control and cognitive tasks. The motor behavior, called *travel*, is the viewpoint's movement from one location to another. The cognitive process, called *wayfinding*, is the specification of a path through an environment. During traveling, wayfinding is supported mainly by regular rotations of the view to scan the environment passing by. For that purpose, the rotational DOF must be controlled independently of other input channels. GlobeFish's tangible separation of rotational input from translational input facilitates this environment scanning and thus wayfinding. Different interaction metaphors may support travel along a given path.

The Groovepad

This device consists of a regular touchpad surrounded by an elastically suspended ring with joystick-like functionality. Its two input sensors can be used separately but facilitate frequent and fluid switching between their different input characteristics. In addition, the Groovepad can contain a tracking sensor to facilitate pointing and selection.

We originally developed the Groovepad for the Two-4-Six, a hand-held input device used to navigate in 3D-graphics applications.[4] In this case, the touchpad specified position-controlled view orientation in a virtual world, while the elastic ring provided rate-controlled viewpoint motion. To rotate the viewing direction around the vertical axis, the user moved a finger along the elastic ring (see Figure 1b). A separate analog rocker sensor operated by the index finger controlled movement in the depth dimension. This sensor configuration worked well for basic navigation in virtual buildings such as museums or exhibition halls.

Considering 2D desktop applications, we found that the Groovepad matches the required functionality for the increasingly popular zoomable interfaces quite well. The elastically suspended ring can serve as a tangible correspondence to the application window's frame. It's used for panning the workspace, while the touchpad is used for pointing inside the window. Smooth circular gestures along the Groovepad ring specify the zoom factor. An initial user study showed that users performed better with the Groovepad than with regular touchpad interfaces. This was particularly true for tasks requiring frequent switches between panning the window and controlling the mouse pointer.

The Unexplored Design Space

Our user studies indicate that these devices perform well for a certain set of tasks and that they can compete with commercially available solutions. To a great extent, however, the design space for desktop as well as handheld devices remains unexplored. To further improve 3D UIs, further user studies based on carefully selected tasks and task combinations must examine the advantages and disadvantages of various sensor combinations.

The Biosignal Interface as a Novel 3D UI

An important goal of VR technology is to allow intuitive interaction between the user and the virtual world. For example, in VR technology development's early days, a user wearing a data glove with a Polhemus sensor could grasp and pick up a virtual 3D object. Now, however, we can imagine a much wider range of interactive channels. Here, we describe the use of biosignals as an input channel for 3D UIs.

The biosignal channel provides a different way for users to interact with a virtual environment, sometimes without any physical motion. In addition, we can directly measure the user's invisible states, such as intention.

The relatively new research field of brain-computer interfaces fits this category. Researchers have reported a variety of brain activities that can serve as an interface channel, such as visual evoked potential (VEP),[5] p300 evoked potential,[6] and motor imagery.[7]

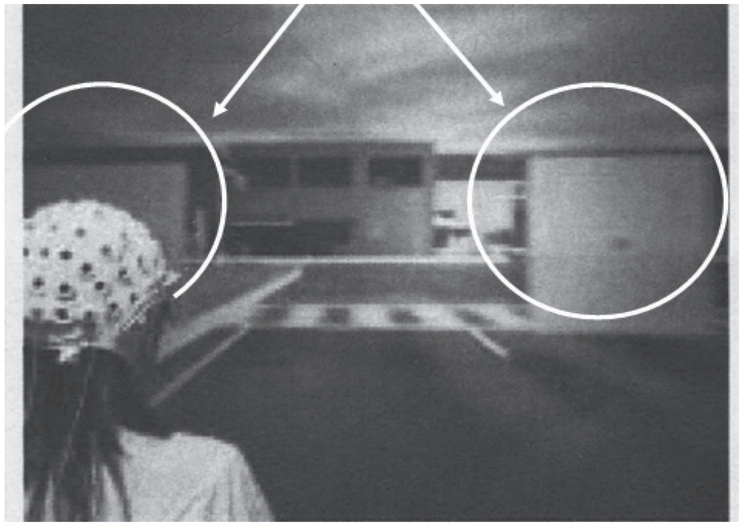


Figure 2. Developing biosignal-based interfaces at the University of Tokyo: A test subject mentally selects the button on the left or right by staring at it. The interface determines the selected button by measuring the subject's EEG signals.

Steady State VEP

A VEP is a measurable signal that arises owing to the stimulation of the visual cortex. In particular, steady state VEP (SSVEP), which occurs in the visual cortex when the subject views a flickering stimulus with a frequency of more than 4Hz, is known to be a reliable signal.

Our research laboratory has performed several experiments using SSVEP as a virtual joystick to navigate a 3D virtual environment displayed in the Cabin (Computer Augmented Booth for Image Navigation), a VR room with a five-screen configuration.[8,9] As Figure 2 shows, we positioned virtual buttons to select left and right turns in the virtual environment at a distance of 2.0 meters from the user and a view angle of approximately 13 degrees. We set the flickering frequency to 8.0 Hz for the left-turn button and 6.0 Hz for the right-turn button. We requested the subject to look at the left button if he or she intended to turn left, and vice versa.

We used a modular EEG cap system to measure the EEG signal on the subject's scalp. To generate the control signal, we used three-channel EEG signals: PO7, PO8, and Oz. We extracted the EEG features from a linear combination of the voltages of the three signals as $V_{Oz} - (V_{PO7} + V_{PO8})/2$. Using support vector machines (SVMs) with a linear kernel, we classified two states of brain activity—whether the subjects were focusing on the left

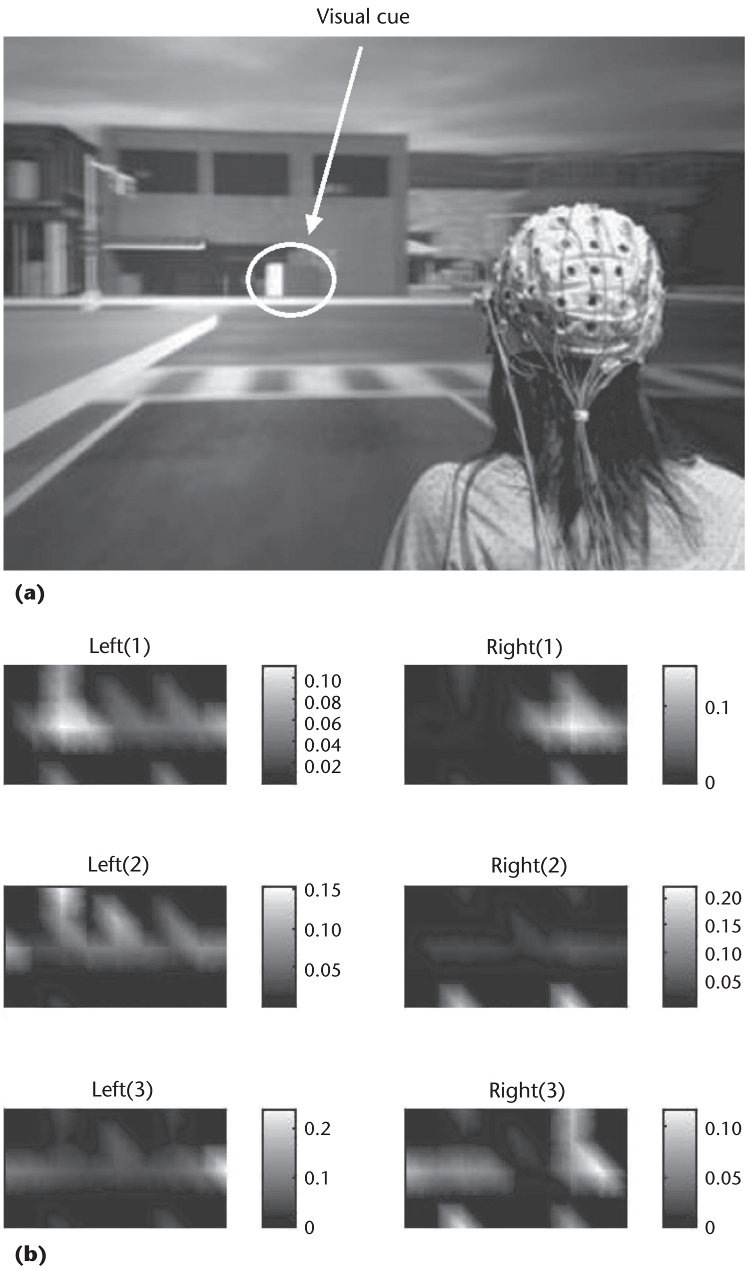


Figure 3. Using motor imagery as an interface: (a) In response to a cue, test subjects imagined tapping their left or right finger. (b) Common-spatial-pattern filtering of the motor imagery tasks revealed three significant components (1–3) of the EEG pattern map.

button or the right button—with approximately 70 to 80 percent success.

Motor Imagery

A motor imagery signal that's fully recognized and classified can serve as an interface that reacts simply to thought. For example, the mu-rhythm is an EEG component that has 8 to 12 Hz frequency and is typically observed at somatosensory cortices. Body movement suppresses the mu-rhythm; this suppression is called *event-related desynchronization* (ERD). Interestingly, ERD occurs even without actual motor movement. Common spatial patterns (CSPs), which are linear spatial filters, can extract the features of signal patterns such as multichannel EEG after a learning period.[10]

Figure 3a shows our experimental setup. We asked the subjects to produce a motor imagery signal (we asked them to imagine tapping their left or right finger) in response to a cue. Using the EEG cap, we measured 16-channel EEG signals.

To process the EEG signal, after band-pass filtering (8 to 30 Hz) we applied the CSP algorithm and chose an SVM with a linear kernel for classification. For the learning data for the CSP algorithm, we used an EEG signal pattern without visual feedback (the case when motion imagery doesn't cause motion in the virtual environment).

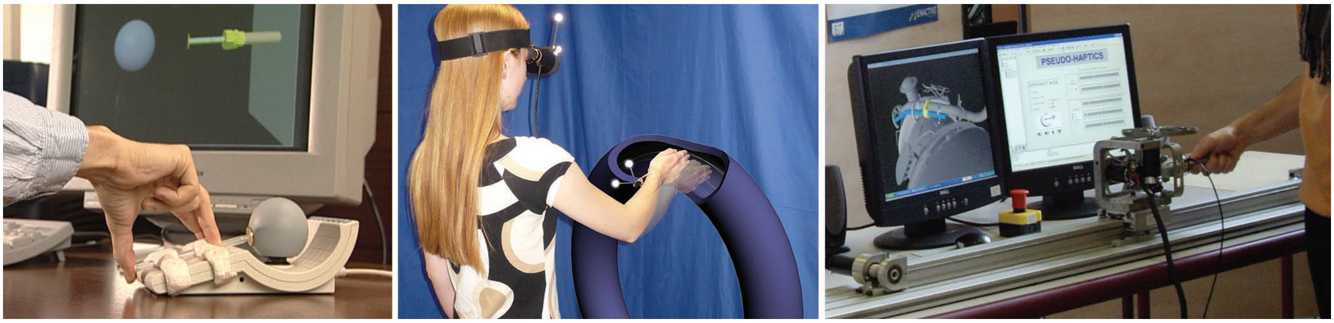
Figure 3b shows the three significant components (1–3) of the EEG pattern map after CSP filtering during phantom finger movements. We produced the EEG pattern by projecting the 16-channel electrode montage onto a rectangular map. As the figure shows, when the subject images left-finger tapping, the hemisphere's left side significantly contributes to the spatial patterns, whereas for right-finger tapping, the right side contributes. This shows that you can successfully extract ERD from motor imagery. On the basis of this result, SVMs could classify left and right commands successfully. We achieved almost 80 percent success.[11]

Although researchers have reported that well-trained subjects can generate localized EEG even without CSP filtering, most people can't. So, CSP filtering is essential.

Pseudo-haptic Interfaces

Three-dimensional UIs increasingly require integrating several input modalities and several types of sensory feedback. Together with visual and auditory sensations, haptics is one of the most important types of sensory feedback. Active force feedback requires the availability of a haptic device able to return forces. However, such devices aren't always available and are often difficult to integrate in VR configurations because of the space taken up by the hardware components. In addition, to guarantee stiff and stable rendering, active haptic rendering requires both complex computations and a refresh rate of approximately 1 KHz for the haptic loop.

Whereas active haptic feedback is necessary for some applications, a growing number of studies focus on alternative, lighter approaches such as sensory substitution,[12] passive haptics,[1] or pseudo-haptics. Pseudo-haptic systems are "systems providing haptic information generated, augmented,



(a) **(b)** **(c)**

Figure 4. Two examples of pseudo-haptics research and an application: (a) A virtual spring's stiffness is pseudo-haptically simulated by coupling a force sensor and visual feedback.[14] (b) With HEMP (Hand-Displacement-Based Pseudo-haptics), the user experiences a simulated force field.[17] (c) In Haptic Wrists, the hand-roll degree of freedom (DOF) along the hand axis is pseudo-haptically controlled, while the two other DOF are haptically controlled.[18] Figure 4c used with permission of Javier Martin and his colleagues.

or modified by the influence of another sensory modality."[13] Here we discuss some of the most recent pseudo-haptics research and applications.

Pseudo-haptic Simulation

Several researchers have proposed pseudo-haptic solutions based on visuohaptic illusions. These methods exploit the visual modality's domination over the haptic modality. Basically, these methods perturb the visual feedback of the hand representation (the cursor, for instance), or the representation of an object handled by the hand, to induce the haptic sensation that would cause this perturbation.

At INRIA, we've proposed a simple example in which a mouse controls a cube's displacement on a horizontal plane.[14] A qualitative experiment showed that test subjects could perceive perturbation of the speed of the cube's displacement as friction-like (friction, gravity, or viscosity) feedback. Likewise, Anatole Lécuyer and his colleagues[15] and Koert van Mensvoort[16] showed that a similar setup with a mouse controlling a cursor can produce sensations interpreted as elevations or material properties.

Although our test subjects experienced a sense of friction, gravity, or viscosity, we noticed that forces are more perceptible with a force-sensor-based input device (for example, a Spaceball) than with a mouse.[14] We hypothesized that the reaction force from the force sensor is more perceptible. This observation led us to propose a second class of pseudo-haptic solutions, based on force sensors. The first studies concerned simulating stiffness. We proposed a pseudo-haptic virtual spring based on the coupling of a force sensor and a perturbed visual feedback.[14] We deduce the virtual spring's displacement (D_{virtual}) from the force applied by the user (F_{user}) and the virtual spring's stiffness (K_{virtual}), using Hooke's law:

$$F_{\text{user}} = K_{\text{virtual}} \times D_{\text{virtual}} \tag{1}$$

A major advantage of the force-sensor-based pseudo-haptic approach is its relevance to real parameters. A quantitative evaluation of this setup for compliance discrimination between a real and a virtual spring shows that subjects can discriminate successfully with a just noticeable difference consistent with previous studies on manual compliance discrimination (see Figure 4a).

We've observed similar results with torques. In 2004, we extended the concept of pseudo-haptic feedback of stiffness to torque feedback.[19] Torque pseudo-haptic feedback is based on the coupling of visual feedback and the internal resistance of a force/torque sensor that passively reacts to the user's applied force. Results showed that our experiments successfully simulated torque pseudo-haptic feedback, with different performance for isometric and elastic input devices. We also detected this difference when simulating pseudo-haptic stiffness. In another study, Lionel Dominjon and his colleagues showed that perturbation of the visual feedback could modify mass or weight perception.[20]

More recently, we introduced a deviceless pseudo-haptic concept based on an AR setup employing a video see-through head-mounted display (HMD). HEMP (Hand-Displacement-Based Pseudo-haptics) aims to simulate haptic sensations by decoupling the hand's visual feedback from its proprioceptive position.[17] We applied HEMP to the simulation of force fields; the user experiences the simulated force field by slowly moving a hand along the flow (see Figure 4b). Initial experiments showed that the subjects perceived a sensation of force.

Applications

Our first applications employed a mouse, whereas later applications employed the force/torque sensor. Developing mouse-based pseudo-haptic applications is relatively simple. A pseudo-haptic mouse can, for instance, be used for map navigation. To help the user roughly locate a landmark while panning a map, we make the mouse pointer lighter (faster) as the mouse approaches the landmark.[21]

Such a technique can also enhance GUIs. Regan Mandryk and her colleagues created "sticky widgets" to ease users' access to some interface elements.[22] The mouse slows near these widgets, creating the illusion of stickiness. User studies compared several levels of stickiness and demonstrated this approach's benefits. Several other examples such as games or the simulation of elevations appear elsewhere.[16]

Several promising applications are based on the principle that forces are more perceptible with a force sensor than with a simple position sensor such as a mouse. Franck Crison and his colleagues presented VTT (Virtual Technical Trainer), a pseudo-haptic trainer for metal-milling machines.[23] VTT's pseudo-haptic interface is a Spaceball. The correlation between the user's force on the Spaceball to move the tool and this move's speed (depending on the metal's resistance) as seen on the visual display generates the pseudo-haptic perception.

Combining pseudo-haptic solutions with actual haptic systems would also be interesting. CEIT (Centro de Estudios e Investigaciones Técnicas de Gipuzkoa) developed a smart 3-DOF haptic wrist by combining real haptic DOF and pseudo-haptic DOF (see Figure 4c).[18] The hand-roll DOF along the hand axis is pseudo-haptically controlled, while the two other DOF are haptically controlled. Based on Alexis Paljic and his colleagues' research,[19] the pseudo-haptic DOF is possible thanks to a force/torque sensor measuring the torque the user exerts along the handle axis. The associated visual feedback lets the system control the pseudotorque sensation returned to the user. Mixing haptics and pseudo-haptics, if they're well integrated, should be a promising solution to avoid some limitations of haptic systems.

Benefits

Pseudo-haptics can be a viable alternative or complement to real haptics in certain situations. It can increase performance and the impression of reality while lowering the cost of haptic systems, owing to lower requirements regarding actuators and computational load. However, it's still a young research area, and many questions remain open.

3D Interfaces for Multidisplay Environments

Offices and meeting rooms often incorporate a variety of display combinations, involving projection screens, wall-sized LCDs, and desktop and notebook PCs. We often use these multiple displays simultaneously during work. Moreover, digital tables are becoming popular in such environments. With the increasing amount of information produced by computers and the decreasing cost of display hardware, such multidisplay environments (MDEs) are becoming increasingly common.

We expect to work effectively by using multiple displays in such environments; however, important problems prevent users from effectively exploiting all the available displays. Displays might be at different locations from and different angles to the user. As a result, managing windows, reading text, and manipulating objects can be difficult.

So, we need a sophisticated 3D interface for MDEs that stitches the displays seamlessly and dynamically according to the users' viewpoints. This will let users interact with the displays as if they're in front of an ordinary desktop GUI.

Problems in MDEs

Ordinary GUI environments are designed with the assumption that the user sits in front of a stationary display perpendicular to his or her view; windows and data are rendered according to this assumption. Unfortunately, the perpendicularity assumption doesn't always hold in recent display environments. When the display plane isn't perpendicular to the viewer (for example, tabletop displays), when the display is flat and covers a large viewing angle (for example, a large display seen from close proximity), or when the user moves around, the viewing angles become more oblique. The violation of the perpendicularity assumption results in increased difficulty in viewing, reading, and manipulating information owing to perspective distortion.[24]

The perspective problem becomes more crucial in MDEs than in single-display environments. Misunderstandings due to perspective distortions might decrease collaboration's efficiency. Moreover, if information extends to multiple displays, part of the information might not be visible and will consequently be difficult to interpret. On the other hand, even in MDEs users expect to use interaction techniques familiar in ordinary GUIs instead of unfamiliar special techniques. So, we need to be able to provide MDE users with techniques that extend the ordinary GUI environment. Two basic techniques that treat multidisplays as part of one large virtual GUI environment are Perspective Cursor and Perspective Windows.

Perspective Cursor

Merriam-Webster OnLine defines perspective as "the appearance to the eye of objects in respect to their relative distance and positions." Perspective Cursor presents a cursor that moves beyond display boundaries seamlessly as if it's in an ordinary desktop environment[25] (see Figure 5a). It calculates the cursor's position and movement from the user's viewpoint on the assumption that the system knows the spatial relationships between the user's viewpoint and the visible displays. So, the user perceives the cursor's movement across displays as continuous, even when the cursor's actual movement in 3D space isn't.

Users control the cursor on a virtual sphere around them. Because the cursor movement mapping is spherical, the cursor might point to areas where there's no display. So, users might lose the cursor in these spaces. The solution is a perspective variant of halos.[26] Halos are circles centered on the cursor that are big enough in radius to appear, at least partially, in at least one screen (the red arc in Figure 5b). By looking at the position and curvature of the displayed part of the circle, users can tell how far and in which direction the perspective cursor is. When the cursor is just off of one display, the displayed arc is highly curved, showing most of the circle. If the cursor is far away, the arc will resemble a straight line.

Perspective Windows

This technique shows perspective-corrected information that users observe seamlessly as if it were perpendicular to them, even if it's spread over several displays[27] (see Figure 5b). Perspective Windows displays the same kind of content as traditional 2D windows (for example, a Web browser or a text processor) but offers extra features derived from its perspective-aware capabilities.

Unlike regular windows, Perspective Windows provides optimal visibility to the user regardless of the display's angle. It renders the windows using a virtual plane that's perpendicular to the user in the center of the window and then projected onto the display. If a window is displayed across more than one surface simultaneously, perspective can help reduce fracture. (see Figure 5c).

The Future of Multidisplay Environments

Perspective Cursor and Perspective Windows provide a perspective-correct GUI for viewing, reading, and manipulating information for each MDE user. Although these techniques consider only the visual displays, future work will likely include other modalities of information presentation, such as audio. Future MDEs will be required to give the appropriate people the appropriate information by satisfactorily combining the available devices according to

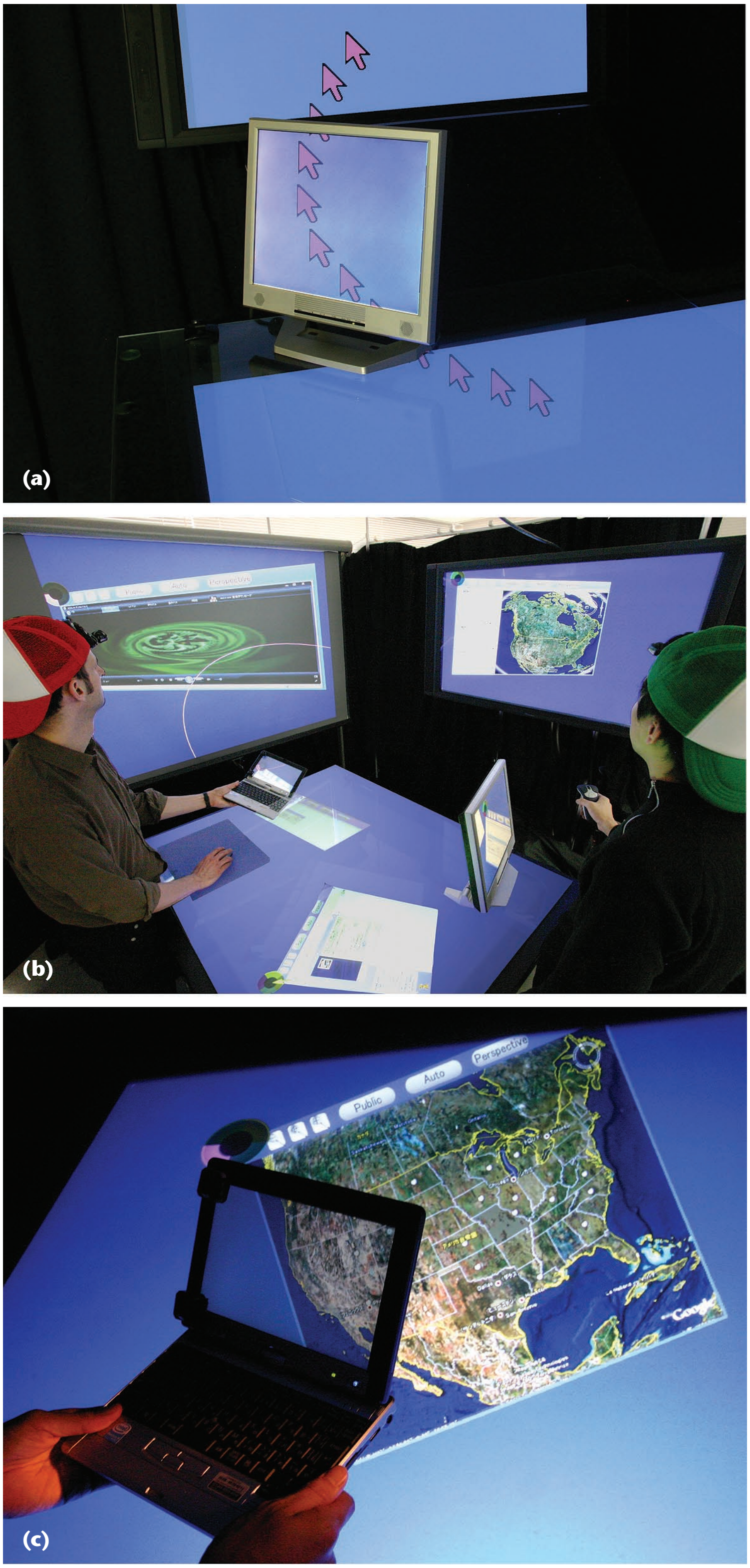


Figure 5. Techniques for 3D user interfaces (3D UIs) in multidisplay environments: (a) Perspective Cursor presents a cursor that moves beyond display boundaries seamlessly as if it's in an ordinary desktop environment. (b) Perspective Windows shows perspective-corrected information that each user observes seamlessly as if it were perpendicular to him or her, where the red arc indicates how far and in which direction the cursor is. (c) Perspective Windows shows perspective-corrected information even if it's spread over several displays.

the situation. This will require adequate sensors; measuring such human dynamics as gestures (including pointing and gaze), physiological information (such as pulse rate and body temperature), and brain waves will be important.

Analyses of the conversations between people sharing locations and communication processes will provide even more data to better align the necessary information. The ideal MDE will identify the situation in the room even if users don't do so explicitly and will provide the necessary information to eligible people in the room. Such a future environment is often called an "ambient information environment." Toward this end, challenging studies of 3D UIs for MDEs will continue.

To leverage humans' experience in the real world, the correct default for any 3D system is for all objects to attach to other objects.

Guidelines for Easy-to-Use 3D Interaction Techniques

Three-dimensional UIs in VR are still in their infancy. Part of the problem is that 3D hardware technologies are still immature and that setting up and keeping a VR system running on a daily basis incurs significant overhead. Another problem is that many user interface techniques are implemented as a thin layer on top of the mathematical foundations. One example is handles that allow movement only in the directions of the three major coordinate axes. Consequently, only users who understand the underlying mathematics can effectively use such a system. The biggest problem is that most VR systems require significant training before they can be used productively, which is the primary barrier to broad acceptance.

In contrast, many 3D games and online virtual worlds offer easy access to 3D content. Most people adapt quickly to the way such systems afford interaction with 3D worlds. Interestingly, most of these systems use 2D input devices for interaction, which involves the additional overhead of understanding the mapping of 2D mouse movements to 3D motions on the screen.

Here are eight guidelines derived from user studies with novice participants (that is, persons without VR knowledge); results from VR, perception, kinesiology, and 2D GUI research; and lessons learned from 3D games. These guidelines will help drive 3D UIs toward broader accessibility and will form the basis for the next generation of 3D UI techniques.

Floating Objects Are the Exception

In the real world, few floating objects exist, and almost all objects are attached to other objects. To leverage humans' experience in the real world, the correct default for any 3D system is for all objects to attach to other objects. In this context, it's important to note that all professions performing "full" 3D tasks (astronauts, fighter pilots, divers, and so on) need substantial training to perform their work. Unfortunately, most VR (and CAD) systems use the default of having every object float. In good 3D UIs, users should be able to make individual objects float. But this should be the exception, not the rule! In other words, standard 3D UI techniques can and should be based on object attachment. Interfaces can incorporate special mechanisms to make objects stay in midair.

Objects Don't Interpenetrate

Solid objects—including the viewers themselves—can't interpenetrate each other. Humans are used to this and deal with it every day. However, many VR systems allow object interpenetration by default. Interpenetration leads to confusing visual display, and many novice users can't easily recover from such situations. For example, consider the negative effect of users being "trapped" behind a wall in a game—most novices need help to recover from such a situation. Today, performing real-time collision detection and avoidance for large environments is easy—for example, with the help of graphics hardware.[28] As an added benefit, collision detection and avoidance enables sliding contact, an efficient way to position objects in the real world.[29]

Interaction Should Be Only with Visible Objects

Strong evidence exists that users prefer to navigate to manipulate occluded objects.[30] This has several consequences. First, it points to the importance of easy navigation. Second, because a 2D manifold can fully describe the set of all visible objects, 2D input is sufficient to select an object. This is also documented by the success of ray-casting-based techniques relative to full 3D selection techniques.[31] This also means that 2D input devices are sufficient to select objects in a 3D world—assuming that adequate 3D navigation techniques exist.

Perspective and Occlusion Are the Strongest Depth Cues

For manipulation of objects beyond arm's length, perspective and occlusion are the strongest depth cues.[32] Assuming that there are no floating objects, these two cues are usually sufficient to accurately and quickly judge objects' 3D position in an environment (unless optical illusions are involved). Although stereo display has a clear value, it matters most for objects fairly close to the viewer. So, stereo display of 3D environments isn't always necessary. Last, but not least, evidence exists that most stereo technologies are far from mature and are tiresome or problematic if used daily.[33]

People See the Object, Not the Cursor

Research into primate vision has demonstrated that monkeys attend visually to not only the tip of a tool in their hand but also the whole tool and the hand. This indicates that a cursor might not be the best choice for 3D UIs—a cursor is effectively a point, while an object covers an area in the visual field. The Sesame (Sketch, Extrude, Sculpt, and Manipulate Easily) sliding technique analyzes the visual-area overlap between the manipulated object and the static scene to determine a moving object's position.[34] The user studies reported in conjunction with this research demonstrate that users can easily use and learn such techniques and that such methods provide clear performance benefits.

Full 3D Rotations Aren't Always Necessary

Many common objects, such as chairs, desks, and shelves, have a clear "up" orientation. Other objects, such as hanging lamps and whiteboards, also have clear orientations. These objects are all attached to other objects. This attachment also provides constraints for rotation—a chair is on its side only in exceptional cases. Consequently, providing a simple user interface to rotate an object around the axis afforded by that object's main attachment is a good design alternative for simple-to-use systems.[35] Although the interface should support full 3D rotations, such modes can be secondary and don't need to be easily accessible.

2D Tasks Are Cognitively Simpler Than 3D

Most real-world tasks aren't fully 3D; they're 2D or 2-1/2D. For example, buildings consist of layers of 2D floor plans because they're easier to build and navigate that way. Real 3D structures in buildings exist, but they're the exception, not the rule. Consequently, most humans are used to dealing with 2D or 2-1/2D and don't have the training necessary to deal with fully 3D problems. For 3D UIs, this means that offering 2D alternatives for many tasks is an excellent way to increase usability.

2D Input Devices Are Advantageous

Input devices such as the Personal Interaction Panel, which use a pen on a 2D tablet to provide interaction in a VR system, have been shown effective for 3D worlds.[36] Also, constraining the input to 2D combats hand fatigue and provides more accuracy.

Moreover, a comparison of input device specifications between mouse- or pen-based systems and 3D technologies reveals that 2D technologies are one to two orders of magnitude more precise.[37] This research also shows initial evidence that this precision is mainly why 2D input devices outperform 3D technologies. Interestingly, a supporting surface's effect is much less than that of increased resolution. Consequently, combinations such as using a tablet PC with accurate pen tracking along with a 3D tracking system for off-slate interaction are a sensible approach.

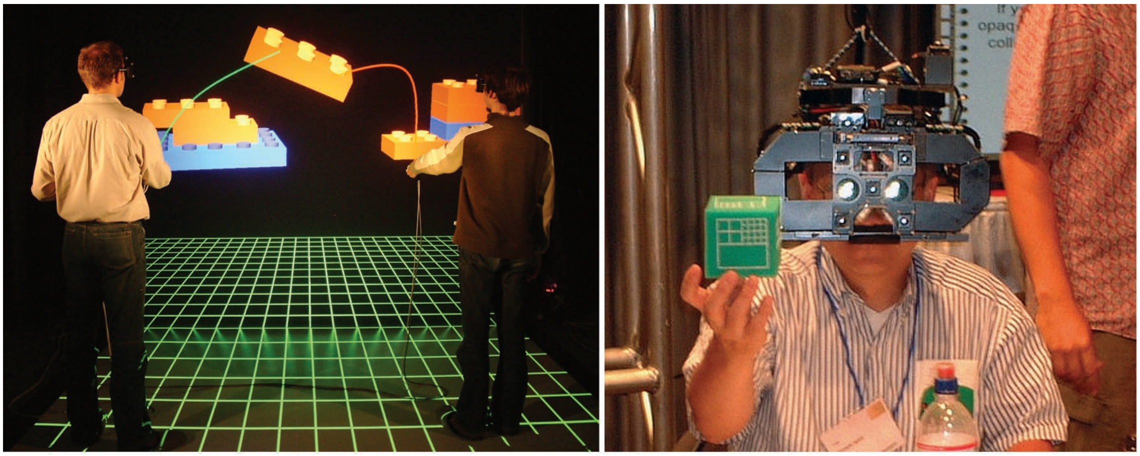
The next generation of 3D UIs can greatly benefit from user interface techniques that are adapted to how humans perceive and interact with the real world.

The Goals

The next generation of 3D UIs can greatly benefit from user interface techniques that are adapted to how humans perceive and interact with the real world. Moreover, novel 3D UIs should leverage the strengths of existing technologies (for both input and output) as far as possible and avoid known weaknesses. This will maximize the chances for skill transfer, thus increasing the developed techniques' usability. This will lead to better 3D applications, a broader range of applications that use 3D productively, and increased adoption of 3D UIs.

Design Considerations for Collaborative 3D UIs

A collaborative 3D UI must accommodate many challenges inherent in its multiuser nature. These challenges include giving correct perspective views to every participant, handling multiuser object manipulation, and supporting natural awareness. To design a successful collaborative system, we must appreciate recent trends, trade-offs, and limitations related to interaction techniques and display technologies.



(a) (b)

Figure 6. Two attempts at collaborative 3D UIs: (a) The Bent Pick Ray renders a straight line (a pick ray) from the input device to the selected object when one user grabs the object and renders a bent pick ray when the second user grabs that object.[40] (b) Kiyokawa's occlusive head-mounted display achieves both occlusion handling and natural visibility of colocated participants.[45]

Multiuser Display Systems

One requirement for a collaborative 3D UI is that participants share the same virtual environment while observing it from their own perspectives. This basic requirement becomes even more difficult when participants are at the same physical location. A conventional stereo projection system can't provide correct perspective views to more than one user. Viewing the scene from a position different from the intended rendering position will skew and distort the user's view. In such an environment, different colocated users will perceive the same virtual object as if it were at several different locations.

Multiviewpoint images composed of different image elements from different viewpoints partially solve this problem.[38] This technique projects interaction elements for each user in the correct position on the basis of the tracked real input devices from the user's viewpoint. Bernd Froehlich and his colleagues have developed a more complete solution by multiplexing the cycle of stereo shuttering.[39,40]

Oliver Bimber and his colleagues[41] and Yoshifumi Kitamura and his colleagues[42] have developed other types of display systems for colocated 3D collaboration. These systems provide stereoscopic images at the same physical location from different perspectives by separating viewing frustums using mirrors[41] or masking plates.[42] One limitation of these types of display is that the effective working volume is fixed and relatively small.

Another common display device for colocated collaboration is an HMD. With a head-tracking facility, virtual objects can be at arbitrary locations. An HMD's typical disadvantages include more noticeable system latency, limited field of view (FOV) and peripheral vision, and ergonomic discomfort.

Multiuser Object Manipulation

Another requirement for a collaborative 3D UI is that participants can manipulate virtual objects cooperatively. Most collaborative systems support simultaneous manipulation of different objects by different users.[43] However, generally, only one user at a time can manipulate a virtual object.

For more flexible collaborative activity, some recent studies attempt to let multiple users simultaneously manipulate an object. Roy Ruddle and his colleagues propose several solutions to combine two users' movements to obtain the virtual object's final movement.[44] Kai Riege and his colleagues describe the Bent Pick Ray (see Figure 6a).[40] This technique renders a straight line (a pick ray) from the input device to the selected object for single-user manipulation. It renders a bent pick ray when the second user grabs the same object. The bent pick ray is still emitted from the input device, tangentially to the pointing direction, and touches the selected object at the intersection point.

Theirry Duval and his colleagues developed SkeweR, a technique that lets multiple users simultaneously grab any part of a virtual object.[46] To determine the grabbed object's translation and rotation, SkeweR considers the positions of those "crushing points." Márcio Pinho and his colleagues propose splitting the task's DOF among the users.[47] Evaluations showed that this technique could be useful when users have two complementary views.

Level of Immersion

The ability to immerse participants in a virtual environment is a major asset of VR. The immersion level significantly affects the participants' sense of copresence and facilitates mutual understanding of the shared environment. Ilona Heldal and her colleagues demonstrated that people working in distributed immersive environments felt more present and were more effective than those in nonimmersive settings.[48] The performance results for the best immersive setting were close to those for a face-to-face setting with real objects.

Immersive projection technologies (IPTs) commonly provide a high level of immersion. However, building an IPT system that supports multiuser observation is costly. In comparison to IPTs, an HMD generally provides a lower level of immersion owing to limited FOV. Although some closed HMDs provide a wide horizontal FOV larger than 180 degrees, the typical horizontal FOV of a see-through HMD remains 60 degrees or less. Kiyoshi Kiyokawa recently developed a projective HMD that could provide over 180 degrees of horizontal FOV as well as see-through capability.[49] Using this display, he and his colleagues are developing an immersive colocated collaboration system.

Face-to-Face Arrangement

Colocated collaboration with a wall-type display forces participants to observe the virtual environment from the same direction. In this case, they have difficulty seeing both the shared information on the screen and other participants at the same time. The space-multiplexed stereo displays previously mentioned support a face-to-face arrangement. A face-to-face setup supports natural communication among participants because it supports nonverbal communication cues such as facial expressions, poses, gestures, and viewing directions.[50]

Kiyokawa and his colleagues found that using an optical see-through HMD to display shared information in the space between participants can improve collaboration efficiency.[51] Their results indicate that the participants' communication was more natural, more social, and easier than with conventional wall-screen and tabletop configurations.[52]

Heterogeneous Perspectives

A virtual environment supports free control over users' viewpoints and scaling factors in their individual reference frames. This flexibility benefits collaborative activities too, enabling configurations impossible in the real world, such as shared perspectives[53] and multiscale perspectives.[54] However, in a standard colocated situation that supports direct visibility of other participants, such flexible perspective settings aren't applicable. In this sense, collaborative VR and collaborative AR are complementary in terms of flexibility and awareness support. For this reason, several collaborative 3D systems support both AR and VR workspaces, and transition between them to maximize user flexibility.[43,55,56]

3D collaboration involving a shared real object is difficult to support with independent observation among participants because any motion of the shared object will affect every participant's view. Shun Yamamoto and his colleagues partially address this problem by using replica objects; their approach allows independent observation of the "same" tangible entity.[57]

Occlusion Handling

True occlusion capability not only enhances visual realism but also improves the collaboration experience. For example, when a user points at a virtual object, we expect his or her finger to occlude the object. Also, a virtual object that appears between multiple users should occlude the users behind it. With projection-based systems, handling such occlusion phenomena is difficult.

If the actual depth information of a real scene is available, a video see-through HMD is suitable for handling occlusion, at the expense of degraded real-world visibility. Closer virtual objects are simply overlaid onto the real image, whereas virtual objects farther away are left unrendered. Normal optical see-through HMDs, however, can't handle occlusion phenomena owing to the nature of optical combiners. Kiyokawa's optical see-through HMD tackles this problem and achieves both occlusion handling and natural visibility of colocated participants[45] (see Figure 6b).

A Continuing Quest

Because of recent technology advancements and new interaction techniques, the design domain of a collaborative 3D UI is rapidly expanding. However, no single configuration is right for all conditions. We hope the issues discussed here give some insight into finding an appropriate design for the target application.

3D UIs for and beyond VR

Traditionally, 3D UI research has been tightly connected to VR. Immersive VR systems such as the CAVE (Cave Automatic Virtual Environment) inherently need 3D UIs because they display 3D visual environments, use nontraditional input devices, and require users to stand up and walk around. Even though the definition of 3D interaction—human-computer interaction in a 3D spatial context[58]—doesn't mention VR, most 3D UI research has assumed the use of VR technology. Almost all the common 3D interaction techniques for tasks such as navigation, selection, and manipulation were designed and developed in the context of VR systems.[58]

What impact has this extensive research had? Has it resulted in widespread adoption of complex, but usable, 3D UIs for VR applications? Frankly, the impact has been smaller than you might expect. Although researchers have successfully demonstrated 3D UIs with high levels of functionality and usability, most real-world VR applications have simple interfaces. At least for now, the results of years of 3D UI research aren't highly visible in the VR world.

Here are two strategies, or proposed research directions, that could increase 3D UI research's impact both within and beyond VR.

Demonstrating 3D Interaction's Benefits

Why don't we see more examples of complex 3D UIs in VR applications? After an initial period of unrealistic hype and expectations in the 1990s, it became clear that immersive VR technology and knowledge weren't yet mature enough for most real-world applications. A few applications that depend on a unique user experience and that don't require ultrarealistic visual imagery, such as entertainment, phobia therapy, and vehicle simulation, became highly successful.[59] However, envisioned immersive VR applications with more stringent requirements, such as for architectural design, classroom education, and military command and control, remained research prototypes in most cases.

But this situation might be changing. As VR technology and our understanding of how to use it have matured, there has been a resurgence of interest in immersive VR. The US National Academy of Engineering recently named "Enhance Virtual Reality" as one of engineering's top 14 grand challenges in the 21st century (www.engineeringchallenges.org/cms/8996/9140.aspx). Many industries are again exploring using immersive VR for productive work, and even consumer applications such as immersive gaming are possible.

So, the opportunity again exists for 3D UI research to have a major impact on complex real-world VR applications. But there's a twist: many new applications use "semi-immersive" technologies such as single large stereoscopic projection screens or tracked handheld displays rather than traditional VR technologies such as HMDs and CAVEs. In these semi-immersive systems, developers often can choose between 3D UIs and desktop-style UIs that use 2D input devices and standard 2D widgets. Desktop-style interfaces are attractive because of their broad familiarity and availability.

If 3D UI research is to make an impact on these "new VR" applications, the research community must provide evidence that 3D UIs have significant advantages over desktop-style interaction. This is the first proposed research direction for 3D UIs: demonstrating 3D interaction's benefits. Although we now know a great deal about how to design efficient, usable 3D UIs, we don't know enough about when to employ them. Compared to other types of UIs, what benefits do 3D UIs provide in terms of user productivity, accuracy, or satisfaction?

To answer these questions, we obviously need empirical studies that compare 3D interaction to other interaction styles. Designing such experiments is difficult. Simply comparing two existing interfaces doesn't provide generalizable results, but crafting similar 3D and 2D interfaces from scratch might result in bias toward one or the other. In research on this topic, researchers at Virginia Tech's Center for Human-Computer Interaction have taken a controlled approach, using the same display, environment, and task for each interface, and working to optimize each interface's design before comparing the interfaces. We found that for distant 6-DOF manipulation tasks, 3D interaction techniques were significantly faster and more accurate than comparable desktop-style techniques.[60] Further studies of this type are needed.

Expanding 3D UI Research's Focus

As noted earlier, the definition of 3D interaction isn't limited to interaction with VR. But research on 3D UIs for non-VR systems has been limited. So, the second proposed research direction is to expand the focus of 3D UI research—to actively look for areas where interaction is problematic and determine whether 3D UIs are a good fit.

The most obvious example comes from the gaming world. Typical gaming systems use TVs or computer monitors to display 2D or 3D graphics and use handheld controllers, mice, or keyboards for input. But the wildly successful Nintendo Wii has showed that games aren't limited to this type of interaction; indeed, adding spatial interaction can dramatically change and improve the gameplay experience. Clearly, the opportunity exists for 3D UI research to help determine what forms of 3D interaction are best suited for gamers.

Another opportunity relates to the growing number of large displays in public places (for example, airport information displays), visualization centers (for example, high-resolution display walls), and homes (for example, home theater setups). Often, these displays are passive, simply providing visual information to people near the display. But in some instances, users desire or need to interact with the information on the display. In an airport, for example, travelers might wish to get more detailed information about their flights. In a visualization center, analysts must zoom, pan, rotate, query, cluster, or annotate the visualized data sets. At home, users need to control all their multimedia devices.

In addition, most of these large displays are remote, meaning that the user can't easily walk up and touch them. These displays also must support users who are walking past them, sitting on a sofa, or standing in front of them. These characteristics make desktop-style or touch-screen interfaces impractical.

With spatial input in 3D UIs, users will be able to interact with these large remote displays while standing up and walking around and while distant from the display, without requiring traditional 2D input devices. With computer vision advances, users might not require any input device at all, instead interacting directly with free-hand gestures in empty space. For example, we're investigating the design of spatial gestures for menu selection on remote displays[61] (see Figure 7a) and for precise object selection and manipulation on high-resolution remote displays (see Figure 7b). These examples don't use immersive VR displays; most of them display only 2D data. We believe this is a promising area for 3D UI researchers to explore.

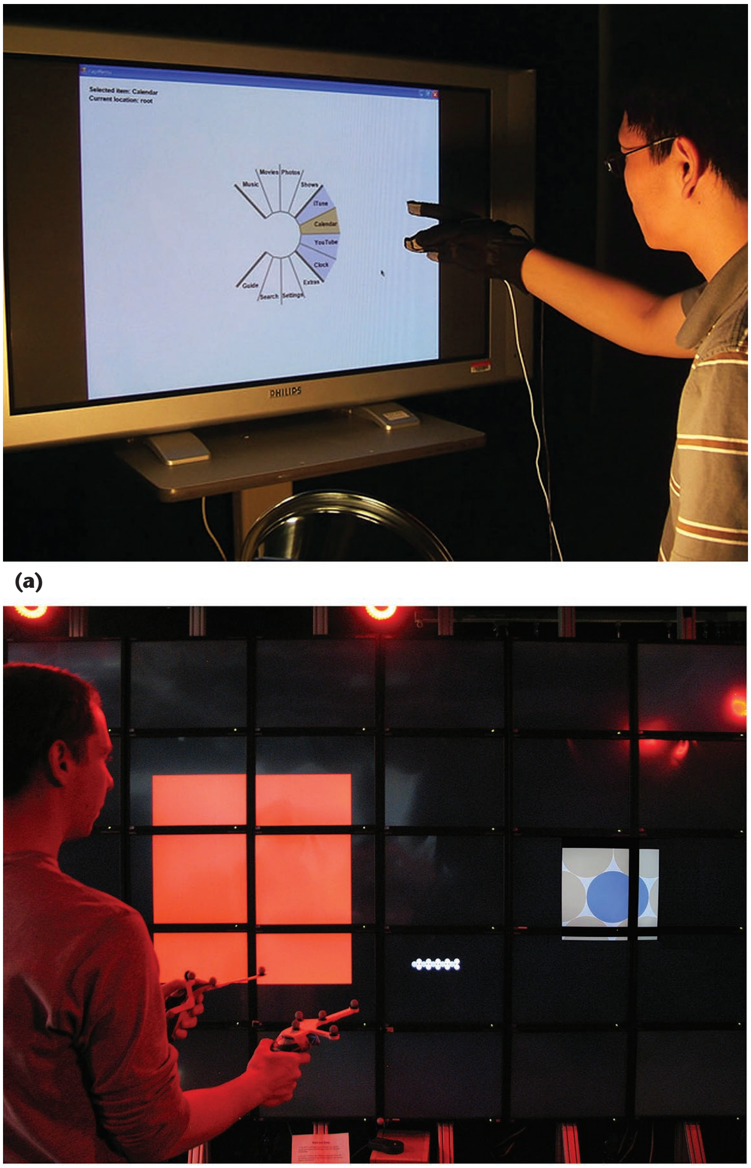
Moving Forward

The 3D UI research community has a broad base of knowledge about the design and evaluation of 3D interaction, but this hasn't yet resulted in a high level of real-world impact. The two strategies described here will change that situation, making that research more relevant to real-world problems.

We hope these pieces serve as an introduction to the field for those new to 3D UIs and as a source of interesting research problems for those already in the field.

Acknowledgments

Bernd Froehlich's section is based on joint research with his colleague Anke Huckauf and his PhD students Alexander Kulik, André Kunert, and Jan Hochstrate. Special thanks go to Alexander Kulik for help with this article. Sabine Coquillart thanks her PhD students (Anatole Lécuyer, Alexis Paljic, and Andreas Pusch) and her colleagues for their contributions to the research presented in this article. The guidelines presented by Wolfgang Stuerzlinger are based on joint research with graduate students Tim Salzman, Graham Smith, Ji-Young Oh, and Rob Teather. Doug Bowman acknowledges the contributions of his graduate students (Ryan McMahan, Regis Kopper, Mara Silva, Tao Ni, and Yi Wang) and his colleagues in the 3D UI community to the ideas in this article.



(a)

(b)

Figure 7. Interaction with large, remote displays: (a) gesture-based menu selection and (b) enhanced 3D pointing for selecting and displaying 2D objects on a large, high-resolution display.

References

1. K. Hinckley et al., "A Survey of Design Issues in Spatial Input," *Proc. 1994 ACM Symp. User Interface Software and Technology*, ACM Press, 1994, pp. 213–222.
2. R. Jacob et al., "Integrality and Separability of Input Devices," *ACM Trans. Computer-Human Interaction*, vol. 1, no. 1, 1994, pp. 3–26.
3. B. Froehlich et al., "The GlobeFish and the GlobeMouse: Two New Six Degree of Freedom Input Devices for Graphics Applications," *Proc. 2006 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2006, pp. 191–199.
4. A. Kulik, R. Blach, and B. Froehlich, "Two-4-Six"—a Handheld Device for 3D-Presentations," *Proc. 2006 IEEE Symp. 3D User Interfaces*, IEEE Press, 2006, pp. 167–170.
5. M. Middendorf et al. "Brain Computer Interfaces

Based on the Steady State Visual Evoked Response," *IEEE Trans. Rehabilitation Eng.*, vol. 8, no. 2, 2008, pp. 211–214.

6. L.A. Farewell and E. Donchin, "Taking off the Top of Your Head: Toward a Mental Prothesis Utilizing Event-Related Brain Potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, 1988, pp. 510–523.
7. G. Pfurtscheller and C. Neuper, "Motor Imagery Activates Primary Sensorimotor Area in Humans," *Neuroscience Letters*, vol. 239, nos. 2–3, 1997, pp. 65–68.
8. H. Touyama, M. Aotsuka, and M. Hirose, "A Pilot Study on Virtual Camera Control via Steady-State VEP in Immersing Virtual Environment," *Proc. Human-Computer Interaction 2008,* ACTA Press, 2008, track 611–065.
9. M. Hirose et al., "Development and Evaluation of the Cabin Immersive Multiscreen Display," *Systems and Computers in Japan*, vol. 30, no. 1, 1999, pp. 13–22.
10. Y. Wang et al. "Common Spatial Pattern Method for Channel Selection in Motor Imagery Based Brain-Computer Interface," *Proc. 27th Ann. Conf. IEEE Eng. in Medicine and Biology Soc.*, vol. 5, 2005, pp. 5392–5395.
11. J. Fujisawa, H. Touyama, and M. Hirose, "EEG Based Navigation of Immersing Virtual Environments Using Common Spatial Patterns," *Proc. 2008 IEEE Int'l Conf. Virtual Reality*, IEEE Press, 2008, pp. 251–252.
12. F. Biocca¸ J. Kim, and Y. Choi, "Visual Touch in Virtual Environments: An Exploratory Study of Presence, Multimodal Interfaces and Cross-modal Sensory Illusions," *Presence*, vol. 10, no. 3, 2001, pp. 247–265.
13. L. Lécuyer et al., "Boundary of Illusion: An Experiment of Sensory Integration with a Pseudo-haptic System," *Proc. 2001 IEEE Int'l Conf. Virtual Reality*, IEEE Press, 2001, pp. 115–122.
14. A. Lécuyer et al., "Pseudo-haptic Feedback: Can Isometric Input Devices Simulate Force Feedback?" *Proc. IEEE Int'l Conf. Virtual Reality*, IEEE Press, 2000, pp. 83–90.
15. A. Lécuyer, J.M. Burkhardt, and L. Etienne, "Feeling Bumps and Holes without a Haptic Interface: The Perception of Pseudo-haptic Textures," *Proc. 2004 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2004, pp. 239–246.
16. K. van Mensvoort, "What You See Is What You Feel: Exploiting the Dominance of the Visual over the Haptic Domain to Simulate Force-Feedback with Cursor Displacements," *Proc. Designing Interactive Systems 2002*, ACM Press, 2002, pp. 345–348.
17. A. Pusch, O. Martin, and S. Coquillart, "HEMP—Hand Displacement-Based Pseudo-haptics: A Study of a Force Field Application," *Proc. 2008 IEEE Symp. 3D User Interfaces*, IEEE Press, 2008, pp. 59–66.
18. J. Martín et al., "Haptic Wrists: An Alternative Design Strategy Based on User Perception," submitted to *J. Computing and Information Science in Eng*; available from jmamezaga@ceit.es.
19. A. Paljic, J.M. Burkhardt, and S. Coquillart, "Evaluation of Pseudo-Haptic Feedback for Simulating Torque: A Comparison between Isometric and Elastic Input Devices," *Proc. 12th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE CS Press, 2004, pp. 216–223.
20. L. Dominjon et al., "Influence of Control/Display Ratio on the Perception of Mass of Manipulated Objects in Virtual Environments," *Proc. 2005 IEEE Int'l Conf. Virtual Reality*, IEEE Press, 2005, pp. 19–25.
21. M. Sugimoto and I. Fujishiro, "Enhancement of Virtual Interaction Devices Using Pseudo Force Sensation," *Siggraph 2000 Sketches & Applications*, ACM Press, 2000, p. 200.
22. R.L. Mandryk, M.E. Rodgers, and K.M. Inkpen, "Sticky Widgets: Pseudo-haptic Widget Enhancements for Multi-monitor Displays," *Proc. 2005 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2005, pp. 1621–1624.
23. F. Crison et al., "Virtual Technical Trainer: Learning How to Use Milling Machines with Multi-sensory Feedback in Virtual Reality," *Proc. IEEE Int'l Conf. Virtual Reality*, IEEE Press, 2005, pp. 139–146.
24. D. Wigdor et al., "Perception of Elementary Graphical Elements in Tabletop and Multi-surface Environments," *Proc. 2007 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2007, pp. 473–482.
25. M.A. Nacenta et al., "Perspective Cursor: Perspective-Based Interaction for Multi-display Environments," *Proc. 2006 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2006, pp. 289–298.
26. P. Baudisch and R. Rosenholtz, "Halo: A Technique for Visualizing Off-Screen Locations," *Proc. 2003 ACM Conf. Human Factors in Computing Systems*, ACM Press, 2003, pp. 481–488.
27. M. Nacenta et al., "E-conic: A Perspective-Aware Interface for Multi-display Environments," *Proc. ACM Symp. User Interface Software and Technology* (UIST 07), ACM Press, 2007, pp. 279–288.
28. N. Govindaraju et al., "Cullide: Interactive Collision Detection between Complex Models in Large Environments Using Graphics Hardware," *Proc. Siggraph Workshop Graphics Hardware*, ACM Press, 2003, pp. 25–32.
29. Y. Kitamura and F. Kishino, "Consolidated Manipulation of Virtual and Real Objects," *Proc. 1997 ACM Symp. Virtual Reality Software and Technology*, ACM Press, 1997, pp. 133–138.

30. C. Ware and K. Lowther, “Selection Using a One-Eyed Cursor in a Fish Tank VR Environment,” *ACM Trans. Computer-Human Interaction*, vol. 4, no. 4, 1997, pp. 309–322.
31. I. Poupyrev et al., “Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques,” *Proc. 1998 ACM Siggraph/Eurographics Workshop Graphics Hardware*, ACM Press, 1998, pp. 41–52.
32. C. Wickens and J. Hollands, *Engineering Psychology and Human Performance*, 3rd ed., Prentice-Hall, 1999.
33. Z. Wartell, L.F. Hodges, and W. Ribarsky, “A Geometric Comparison of Algorithms for Fusion Control in Stereoscopic HTDs,” *IEEE Trans. Visualization & Computer Graphics*, vol. 8, no. 2, 2002, pp. 129–143.
34. J.-Y. Oh and W. Stuerzlinger, “Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments,” *Proc. Graphics Interface 2005*, ACM Press, 2005, pp. 195–202.
35. G. Smith, T. Salzman, and W. Stuerzlinger, “3D Scene Manipulation with 2D Devices and Constraints,” *Proc. Graphics Interface 2001*, ACM Press, 2001, pp. 135–142.
36. Z. Szalavári and M. Gervautz, “The Personal Interaction Panel—a Two-Handed Interface for Augmented Reality,” *Computer Graphics Forum*, vol. 16, no. 3, 1997, pp. 335–346.
37. R. Teather and W. Stuerzlinger, “Assessing the Effects of Orientation and Device on (Constrained) 3D Movement Techniques,” *Proc. 2008 IEEE Symp. 3D User Interfaces*, IEEE Press, 2008, pp. 43–50.
38. A. Simon and S. Scholz, “Multi-viewpoint Images for Multi-user Interaction,” *Proc. 2005 IEEE Virtual Reality Conf.*, IEEE Press, 2005, pp. 107–113.
39. M. Agrawala et al., “The Two-User Responsive Workbench: Support for Collaboration through Individual Views of a Shared Space,” *Proc. Siggraph*, ACM Press, 1997, pp. 327–332.
40. K. Riege et al., “The Bent Pick Ray: An Extended Pointing Technique for Multi-user Interaction,” *Proc. 2006 IEEE Symp. 3D User Interfaces*, IEEE Press, 2006, pp. 63–66.
41. O. Bimber et al., “The Virtual Showcase,” *IEEE Computer Graphics and Applications*, vol. 21, no. 6, 2001, pp. 48–55.
42. Y. Kitamura et al., “Interactive Stereoscopic Display for Three or More Users,” *Proc. Siggraph*, ACM Press, 2001, pp. 231–239.
43. K. Kiyokawa, H. Takemura, and N. Yokoya, “SeamlessDesign for 3D Object Creation,” *IEEE MultiMedia*, vol. 7, no. 1, 2000, pp. 22–33.
44. R. Ruddle, J. Savage, and D. Jones, “Symmetric and Asymmetric Action Integration during Cooperative Object Manipulation in Virtual Environments,” *ACM Trans. Computer-Human Interaction*, vol. 9, no. 4, 2002, pp. 285–308.
45. K. Kiyokawa et al., “An Occlusion-Capable Optical See-Through Head Mount Display for Supporting Colocated Collaboration,” *Proc. 2003 IEEE Int’l Symp. Mixed and Augmented Reality*, IEEE Press, 2003, pp. 133–141.
46. T. Duval, A. Lécuyer, and S. Thomas, “SkeweR: A 3D Interaction Technique for 2-User Collaborative Manipulation of Objects in Virtual Environments,” *Proc. 2006 IEEE Symp. 3D User Interfaces*, IEEE Press, 2006, pp. 71–74.
47. M. Pinho, D. Bowman, and C. Freitas, “Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques,” *Proc. 2002 ACM Symp. Virtual Reality Software and Technology*, ACM Press, 2002, pp. 171–178.
48. I. Heldal et al., “Immersiveness and Symmetry in Copresent Scenarios,” *Proc. 2005 IEEE Virtual Reality Conf.*, IEEE Press, 2005, pp. 171–178.
49. K. Kiyokawa, “A Wide Field-of-View Head Mounted Projective Display Using Hyperbolic Half-Silvered Mirrors,” *Proc. 2007 IEEE Int’l Symp. Mixed and Augmented Reality*, IEEE Press, 2007, pp. 207–210.
50. H. Ishii, M. Kobayashi, and K. Arita, “Iterative Design of Seamless Collaboration Media,” *Comm. ACM*, vol. 37, no. 8, 1994, pp. 83–97.
51. K. Kiyokawa, H. Takemura, and N. Yokoya, “Collaborative Immersive Workspace through a Shared Augmented Environment,” *Proc. SPIE*, vol. 3517, 1998, pp. 2–13.
52. K. Kiyokawa et al., “Communication Behaviors of Colocated Users in Collaborative AR Interfaces,” *Proc. 2002 IEEE Int’l Symp. Mixed and Augmented Reality*, IEEE Press, 2002, pp. 139–148.
53. H. Takemura and F. Kishino, “Cooperative Work Environment Using Virtual Workspace,” *Proc. 1992 ACM Conf. Computer-Supported Cooperative Work*, ACM Press, 1992, pp. 226–232.
54. J. Leigh and A.E. Johnson, “Calvin: An Immersive Design Environment Utilizing Heterogeneous Perspectives,” *Proc. 3rd IEEE Int’l Conf. Multimedia Computing and Systems*, IEEE Press, 1996, pp. 20–23.
55. M. Billinghurst, H. Kato, and I. Poupyrev, “The MagicBook—Moving Seamlessly between Reality and Virtuality,” *IEEE Computer Graphics and Applications*, vol. 21, no. 3, 2001, pp. 6–8.
56. R. Grasset, P. Lamb, and M. Billinghurst, “Evaluation of Mixed-Space Collaboration,” *Proc. 2005 IEEE Int’l Symp. Mixed and Augmented Reality*, IEEE Press, 2005, pp. 90–99.
57. S. Yamamoto et al., “Symmetric Model of Remote Collaborative MR Using Tangible Replicas,” *Proc.*

2008 IEEE Virtual Reality Conf., IEEE Press, 2008, pp. 71–74.
58. D. Bowman et al., *3D User Interfaces: Theory and Practice*, Addison-Wesley, 2005.
59. F. Brooks, "What's Real about Virtual Reality?" *IEEE Computer Graphics and Applications*, vol. 19, no. 6, 1999, pp. 16–27.
60. R. McMahan et al., "Separating the Effects of Level of Immersion and 3D Interaction Techniques," *Proc. ACM Symp. Virtual Reality Software and Technology*, ACM Press, 2006, pp. 108–111.
61. T. Ni, R. McMahan, and D. Bowman, "rapMenu: Remote Menu Selection Using Freehand Gestural Input," *Proc. IEEE Symp. 3D User Interfaces*, IEEE Press, 2008, pp. 55–58.

Doug A. Bowman *is an associate professor of computer science at Virginia Tech. His research interests include 3D user interfaces and the benefits of immersion in VR systems. Bowman received his PhD in computer science from the Georgia Institute of Technology. Contact him at dbowman@vt.edu.*

Sabine Coquillart *is a research director at INRIA (the French National Institute for Research in Computer Science and Control) and LIG (the Laboratory of Informatics of Grenoble). Her research interests include VR and 3D user interfaces. Coquillart received her PhD in computer science from Grenoble University and a research-supervising Habilitation from the University of Paris XI. Contact her at sabine.coquillart@inria.fr.*

Bernd Froehlich *is a full professor with the Media Faculty at Bauhaus-Universität Weimar. His research interests include real-time rendering, visualization, 2D and 3D input devices, 3D interaction techniques, display technology, and support for collaboration in colocated and distributed virtual environments. Froehlich received his PhD in computer science from the Technical University of Braunschweig. Contact him at bernd.froehlich@medien.uni-weimar.de.*

Michitaka Hirose *is a professor at the University of Tokyo's Graduate School of Information Science and Technology. His research interests include human-computer interfaces, interactive computer graphics, wearable computers, and VR. He's a project leader of the Digital Public Art project and the head of Cyber Interface group of the Information and Robot Technology project, sponsored by Japan's Ministry of Education. Hirose received his PhD from the University of Tokyo. He is a member of the ACM and IEEE and is vice president of the Virtual Reality Society of Japan. Contact him at hirose@cyber.rcast.u-tokyo.ac.jp.*

Yoshifumi Kitamura *is an associate professor at Osaka University's Graduate School of Information Science and Technology. His research interests include user interfaces and display systems. Kitamura received his PhD from Osaka University. Contact him at kitamura@ist.osaka-u.ac.jp.*

Kiyoshi Kiyokawa *is an associate professor at Osaka University's Cybermedia Center. His research interests include VR, augmented reality, 3D user interfaces, and computer-supported collaborative work. Kiyokawa received his PhD from the Nara Institute of Science and Technology. Contact him at kiyo@ime.cmc.osaka-u.ac.jp.*

Wolfgang Stuerzlinger *is an associate professor in York University's Department of Computer Science and Engineering and a member of the university's Centre for Vision Research. He focuses particularly on 3D user interfaces and VR systems but also has strong research interests in human-computer interaction, display technologies, and collaborative systems. Stuerzlinger received his doctorate from the Vienna University of Technology. Contact him at wolfgang@cse.yorku.ca.*