Building on Realism and Magic for Designing 3D Interaction Techniques

Alexander Kulik ▪ *Bauhaus-Universität Weimar*

Imagination-based interaction can complement reality-based interaction in the design of 3D user interfaces. This hybrid approach could lead to interface design guidelines that promote higher-level consistency, and thus usability, for a large range of diverse interfaces.

Correspondence to real-world experiences often results in intuitive user interfaces. As a principle for interaction design, it lets users build on existing knowledge and skills when working with computer applications. Many 3D computer applications even simulate elements of 3D reality to provide the means for training or planning. So, they benefit particularly from interaction techniques resembling the respective real-world activities. 3D UIs employ tracking and 3D input devices to enable such interaction in the whole operational environment. For instance, VR systems adapt the displayed view to fit a 3D scene with respect to the user's head position in the real world.

Exploiting the versatile human motor skills for more effective computing is an important motivation for 3D UI design. The complex capabilities of skilled motor operations largely exceed the expressiveness and effectiveness of visually and consciously guided actions. One convincing example is professional musicians' rapidity and accuracy in operating their instruments. However, the established paradigm of direct manipulation[1] considers mainly interaction control based on visual feedback. Direct manipulation ought not to be limited to visual response as expressed by the dictum, "What you see is what you get." In contrast, "What you do is what happens" (see Figure 1) advocates that the design of future interfaces should embrace direct motor interaction. The focus shifts from continuously updated visual representations to embodied interaction with computing as a medium—making 3D interaction a central tool.

Three-dimensional interaction methods don't have to mimic reality. More important, they must be effective, fun, or both in the best case. Technically speaking, the functionality and characteristics of tools we design for computing purposes are limited only by the designer's imagination. In software, virtually no constraint exists on mapping user input to application control—besides the required implementation effort.

A tool's usability, on the other hand, is limited by the user's understanding. Users can only apply those functionalities that they know how to handle. The required skills can always be learned, but for intuitive, consistent interaction techniques, we should consider the users' assumptions about the tools' purposes and working principles. These assumptions are based largely on users' experiences in tangible reality but also on cultural knowledge and skills. The design of interaction tools can remind users of such existing knowledge, thus enabling its application in a changed context. Furthermore, it can guide their imaginations to understand concepts that they haven't experienced before. For instance, users would likely recognize a carpet floating in mid air as a flying vehicle for traveling distances.

The seminal book *3D User Interfaces,* by Doug Bowman and his colleagues, presents a comprehensive look at all aspects of 3D UI research.[3] Here I focus on the conceptual differences of reality-based and computer-augmented techniques. I look at fundamental tasks that occur in interactive 3D

 Published by the IEEE Computer Society

graphics and suggest high-level principles for designing appropriate interaction techniques. I present a variety of design solutions, emphasizing that different tasks require specific interaction utilities.

Therefore, and with respect to user differences, a diversity of interfaces is desirable. A multitude of differing concepts, however, will likely hinder intuitive interaction in terms of transferable knowledge and skills. The key to tackling this problem is to maintain consistency with higher-level design principles. Toward that end, I advocate combining two complementary design frameworks. The first is *reality-based interaction* (RBI), developed by Robert Jacob and his colleagues.[4] The second is *imagination-based interaction* (IBI)—that is, computer-augmented (and therefore "magical") interaction techniques.

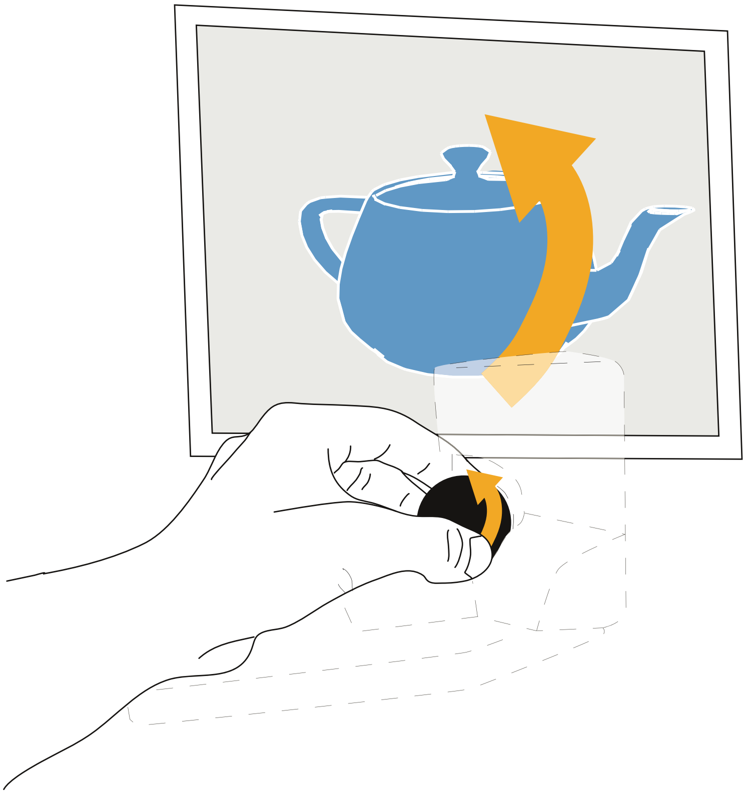

Figure 1. What you do is what happens. Even though users can hardly grasp computer graphics directly, appropriate physical control devices can achieve correspondence of the input activity to the resulting motion on the screen. The Globefish, for example, provides a 3D trackball for rotational input.[2]

Designing 3D Navigation and Manipulation Interfaces

Good interface design deals mainly with equipping users with adequate tools for solving their tasks. Consequently, in *3D User Interfaces: Theory and Practice*, Doug Bowman and his colleagues classified 3D interaction techniques by the intended task.[3] In this article, I focus on the fundamental motion control tasks in interactive 3D computer graphics: viewpoint navigation and object manipulation. They differ in three main ways.

The first is the user's conceptual model. Navigation is self-motion within the virtual environment or, more precisely, a change of the view in the environment. The user's input operations are mapped to the view's motion, so the environment's resulting motion occurs in the opposite direction. This corresponds to walking or driving in the real world. As we move forward, our environment seems to move backward. Objects, in contrast, are manipulated in relation to their surrounding environment. The conceptual model of manipulation thus implies a kinesthetic correspondence between the user's hand motion and the resulting impact on an object.

The second difference is the extent of the interaction space. Viewpoint navigation often requires travelling distances exceeding the interaction system's given operational environment. This inadequacy is a major obstacle for implementing reality-like navigation techniques. Consequently, viewpoint navigation techniques often rely on computer-augmented functionalities such as scaling and automation. Manipulation, in contrast, requires relatively small interaction spaces. Real-world manipulation is constrained by the kinetic limitations of the user's hand-arm system. So, small operational environments are generally sufficient for executing corresponding interaction techniques for manipulating virtual objects.

The third difference is the perceptual cues. Viewpoint motion is guided primarily by the visual feedback provided by real-time computer graphics. When we move in the real world, we also receive vestibular cues, but interactive computer systems can only partially provide these, as I discuss in more detail later. When manipulating objects in the real world, we rely on visual information for planning and verification of the results. During operation, though, our actions are primarily guided by proprioception and haptic feedback from the physical medium at hand. The design of 3D manipulation techniques should take this into account.

Navigation

Depending on the user's objectives, there are three types of navigation: *exploration*, *search*, and *maneuvering* (see Figure 2).[3] The corresponding travel tasks' technical and conceptual requirements imply a preference for specific motion control techniques. So, we can derive important design decisions from a serious task analysis. Note that


(a) (b) (c)

Figure 2. The three types of navigation: (a) exploration, (b) search, and (c) maneuvering. The corresponding travel tasks' technical and conceptual requirements imply a preference for specific motion control techniques.

most applications include several types of navigation tasks. The design of adequate UIs, obviously, must support all the involved tasks—not simultaneously but sequentially.

Exploration. To explore the surrounding environment, we must frequently look around. Humans often and naturally accomplish this by turning their heads. Furthermore, they must often change their direction of locomotion. So, such tasks require steering techniques with support for viewpoint rotations.

Exploratory navigation often involves covering distances without knowing the target destination. While moving, the user is interested in observing or scanning the environment. Rate-control techniques can be advantageous for such tasks because they facilitate the maintenance of a specified motion direction and velocity. Controlling smooth viewpoint locomotion en passant, users can focus on examining the traversed environment.

It isn't by chance that many real-world vehicles are rate controlled rather than position controlled. Although the accelerator doesn't directly control a car's velocity, it feels as if it does because the adjusted acceleration is balanced with opposing forces at certain velocities. For traversing distances, we generally prefer such semiautomatic motion to the effort of controlling all intermediary positions in sequential steps (position control). Position control can be comparably cumbersome and often results in jerky movements.

In contrast, natural walking is a type of position-controlled navigation that promotes fairly fluid motion. 3D computer applications can benefit from corresponding navigation techniques. Specifically, the spatial understanding of complex 3D environments can be improved if both visual and kinesthetic motion cues are matching. However, naturalistic walking isn't easily supported in computer applications, nor is it always the preferred type of locomotion.

Search. If the user is instead searching for a known place, the focus is on finding the most efficient way to get there. Given that the system provides appropriate wayfinding aids, search commonly results in more direct, less curved trajectories. Once the user has specified a target destination (for example, by direct pointing), steering is unnecessary, and the viewpoint can be moved automatically. Navidget is a sophisticated example of such pointing-based navigation.[5] Here, the selection of an object includes the specification of the resulting viewing distance and angle toward that object. This approach minimizes the necessity of further viewpoint maneuvering.

To gain a better overview or select distant targets, elevating the viewpoint or scaling the environment is often desirable (for good examples, see *3D User Interfaces*[3]). Recently, Regis Kopper and his colleagues introduced a 3D navigation interface that exploits pointing and steering techniques to navigate efficiently through multiscale virtual environments.[6]

Maneuvering. Maneuvering describes viewpoint navigation that requires only small-scale motion to arrive at a certain view of the scene. Maneuvering also provides visual depth cues through motion parallax. Here, the user is often focusing on a particular object of interest. Moving around it with unconstrained navigation techniques requires simultaneous translation and rotation. As humans, we've learned to control our physical bodies well enough to effortlessly perform this task. As we continuously turn our heads during walking to observe a particular object, we don't even recognize the complexity of our motion activities.

In contrast, using some kind of vehicle to move around a point of interest while keeping the point in focus is difficult. It's often similarly difficult in virtual environments that don't offer sufficient physical interaction space to directly walk around virtual objects. Fortunately, 3D UIs can provide interaction aids to facilitate such maneuvering. Using bimanual navigation techniques, for example, the user can specify the point of interest with the dominant hand while the nondominant hand induces viewpoint motion around that point.[7]

Another solution is fixed-object manipulation. This technique causes viewpoint motion in the opposite direction because the object is attached to the environment (see Figure 3). Depending on the user's mental model, you could consider this a direct manipulation of the whole environment or indirect viewpoint navigation (as I discuss later).

Because the requirements of maneuvering in relation to objects of interest often correspond to manipulating those objects, we can use similar interaction techniques. In this case, position control is often the preferred transfer function because it corresponds to object manipulation in the real world. Maneuvering tasks needn't involve covering large distances; rather, users often wish to move back and forth between two viewpoints. They can do this more efficiently with position control than with rate control because they can rely on proprioceptive cues about induced motion input.[2]

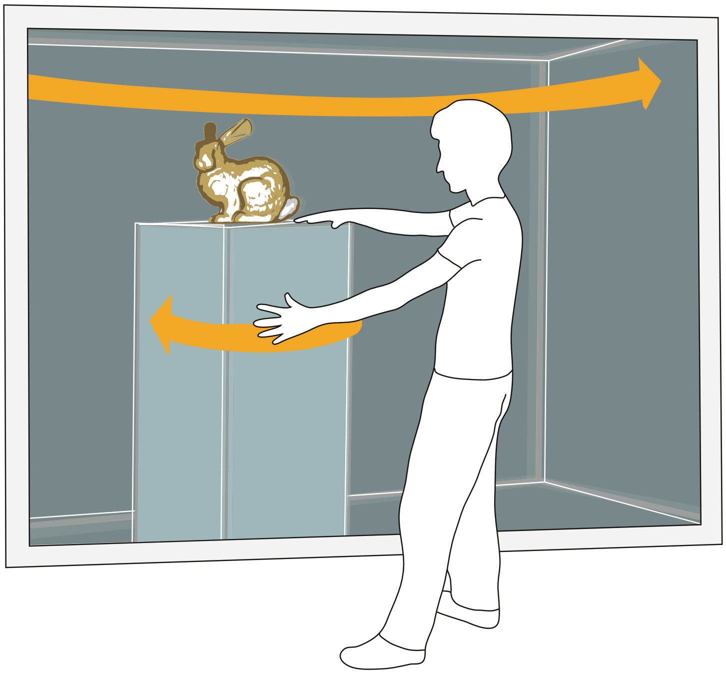

Figure 3. Manipulation-like navigation techniques (for example, fixed-object manipulation) can be an effective way to perform maneuvering tasks. In this case, manipulating the fixed object causes the viewpoint to move in the opposite direction.

Manipulation

In its original sense, "manipulation" refers to changes applied by the human hand. Accordingly, many direct 3D object-manipulation techniques largely correspond to using the real hand (or a tool) to grasp and move virtual artifacts.

Holding a virtual object "in your hands" enables a rich set of direct 3D manipulations. Not only can you rotate and translate it as in reality, but the malleability of computer graphics also facilitates partially "magical" operations such as scaling the object or changing its shape (see Figure 4).

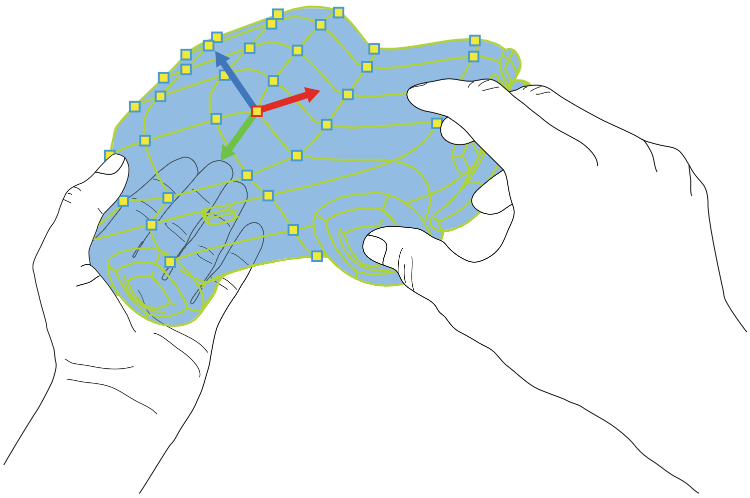

Figure 4. The idea of directly manipulating 3D computer graphics includes grasping an object's surface and deforming it far beyond the possibilities of physical matter.

Manipulation of the environment. In VR, the entire virtual world is as much an object as anything else. So, VR systems sometimes employ manipulation techniques to change the view. For example, a user might grasp an architectural model at some location or even "in the air"[3] to move it around himself or herself. The environment's resulting motion corresponds to the motion direction induced by the user's hand. Such manipulation-based navigation is conceptually like pulling yourself through the environment (navigation) or grasping and moving the weightless virtual world with your hand (manipulation).

Selection and system control. We can view selection as a subtask of manipulation and navigation, used to specify objects of interest or target locations. In most cases, to select something, users manipulate pointing tools such as a cursor or a ray. Researchers have proposed alternatives such as speech and gaze control,[3] but pointing in 3D space is a particularly powerful and intuitive interaction technique. I'll discuss this in more detail later.

System control—the manipulation of nongeometric object parameters—usually involves manipulating 2D menu-like structures embedded in the 3D world. So, users sometimes must use 3D interfaces to interact with 2D structures. This requires manipulation techniques with constrained degrees of freedom (DOF). Providing corresponding tangible controller devices specifically for such frequent subtasks seems effective.[8]

Requirements of Navigation and Manipulation Techniques

Many object manipulation and viewpoint-maneuvering tasks in 3D computer applications correspond largely to real-world interaction. Interaction techniques that build on these common experiences are therefore a good choice for the design of intuitive, effective UIs. This implies using position control, providing haptic feedback of manipulated objects, and including the user's body motion for viewpoint control.

If interaction with the real world were always easy and effective, there would be no reason to shift many tasks to computer-mediated simulations.

However, if interaction with the real world were always easy and effective, there would be no reason to shift many tasks to computer-mediated simulations. In fact, direct interaction with the real world has severe limitations. Consider the irreversible results of actions. Also, the reproduction and adaptation of physical objects is generally cumbersome. Scaling isn't possible. So, fine-grained motion without physical support lacks accuracy, and covering distances is time-consuming. As humans, we have a strong desire to overcome such limitations and use our imaginations to do so. This is why I advocate RBI and IBI as two complementary design principles for 3D UIs.

Reality-Based Interaction

The fundamental approach of VR systems is to at least partially immerse the user into a computer-generated environment, which was and still often is a simulation of some real-world-based scenario. Accordingly, the respective UIs build largely on real-world experiences. Many contemporary and emerging UIs—also outside the VR domain—successfully adapt this idea of reality-based interaction. Robert Jacob and his colleagues' RBI framework covers many emerging interaction styles.[4] It identifies four design themes drawing on references to the real world: *naive physics*, *body awareness and skills*, *environmental awareness and skills*, and *social awareness and skills*. Here, I illustrate this approach by presenting some clever implementations of viewpoint navigation and object manipulation.

Head Tracking and Natural Locomotion

Since Ivan Sutherland's "ultimate display,"[9] head tracking has been the most fundamental interaction technique in virtual environments. Instead of manipulating the view through input devices, users naturally look and walk around to explore the virtual world—just as they would in the real world. For projection-based virtual environments, we can consider the 2D display surface an invisible window between 3D reality and a virtual world. By walking around in front of the display, the user looks from different angles through this window. With head-mounted displays, walking lets the user directly explore the virtual world.

Owing to space and technical limitations, virtual environments can only partially support natural walking over longer distances. One of the most promising, but technically challenging, approaches to overcome these limitations is omnidirectional treadmills. They allow for walking in all directions and have been greatly improved since their introduction.[3] Some of them even simulate various terrains.[10]

If such devices aren't available, walking in place is a simple alternative.[11] Real walking obviously provides a more realistic experience. However, compared to input-device-mediated "flying" metaphors, walking in place results in stronger presence—a stronger feeling of being there.[12]

All navigation techniques based on the walking concept build on body awareness and skills and environmental awareness and skills, but only real walking also provides vestibular cues as in the real world. Nevertheless, visual information more strongly affects our environmental awareness. *Redirected walking* exploits this visual dominance to extend the walkable interaction space.[13] The virtual scene interactively and imperceptibly rotates around the walking user. The system can thereby guide the user through the physical space to prevent him or her from leaving the tracking area or bumping into walls. Researchers have recently combined this technique with the idea of portals connecting one space to another.[14] The space beyond the portal can be at an arbitrary location in the virtual world. This enables traversing virtual environments of infinite extent via natural walking.

Redirection of the user's body orientation can also be combined with walking in place. Many projection-based systems don't provide a 360-degree surrounding view. So, users can't look around simply by turning their bodies or heads. Continuously redirecting the user toward the screen center[13] alleviates this problem. So, naturalistic walking becomes applicable even in the most limited physical-interaction spaces.

3D Pointing

Two-dimensional pointing resembles touching locations on a surface. In 3D space, however, we can also point from a distance, thus covering a larger interaction space with less effort. 3D pointing involves more DOF than its 2D counterpart. The pointing ray's origin and direction must be specified. Owing to body awareness and skills as well as environmental awareness and skills, we can still perform this gesture without much effort. In fact, 3D pointing is a common real-world action to guide others' attention. If we track this gesture, we can use it to select locations and items in 3D computer applications.

Immersive 3D graphics environments often visually assist pointing by displaying a ray that appears to emanate from the physical pointing device. Proprioception of hand and finger postures facilitates approximate pointing, but for high accuracy, such visual feedback is helpful. A real-world analogy is a laser beam looming in fog, but only in virtual environments can someone use such a beam to pick up objects and move them around as if the ray were rigid. Although this interaction technique is more "magical" than reality-based, many users often regard such a *pick ray* as having "natural" functionality.

Pointing from a distance facilitates the acquisition of targets on large display walls—also in the context of 2D graphics.[15] However, owing to the pointing ray's levering effect, involuntary tremors of the pointing hand in mid air are equally amplified. Distant pointing thus often lacks accuracy. We can compensate for this by moving closer to the target because accurate pointing is not only easier but also more sensible in proximity where you can see details. Distance instead provides an overview; thus, larger areas might be more relevant for selection.

Sarah Peck and her colleagues recently developed a multiscale pointing technique for large high-resolution 2D displays that takes this relation into account.[15] Their technique uses a cone instead of a ray for pointing. So, the selection area increases with distance, as does the semantic level of selection.

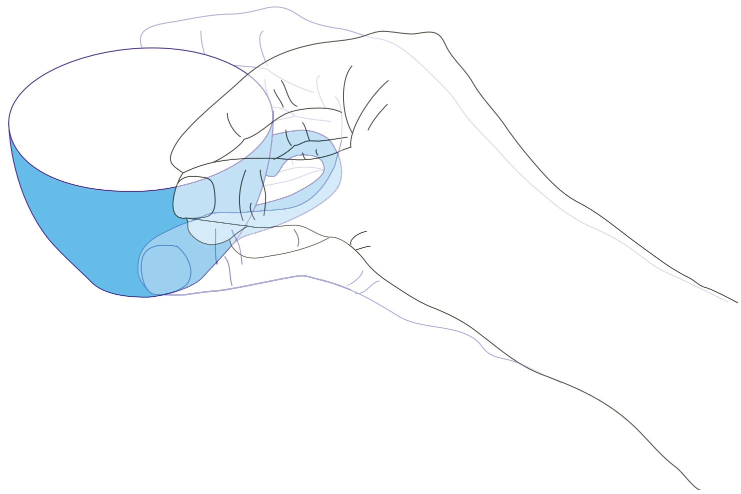

Figure 5. Direct manipulation of 3D computer graphics can be very irritating if you don't expect grasping at nothing.

Direct Manipulation

To manipulate objects in the real world, we grasp them with our fingers and move them around with our hands. To some extent this is also possible in computer applications. Data gloves enable the tracking of hand and finger motions that can be directly mapped to a corresponding representation in a virtual 3D scene.

Unfortunately, technical limitations inhibit natural manipulation techniques from entirely imitating their real-world counterparts. The biggest issue is that graphical objects aren't physically graspable (see Figure 5). So, important haptic information that guides the hand's actions in the real world is generally missing in virtual worlds—unless haptic-feedback devices are used. Currently, haptic input devices are rarely used because they're mechanically complex and user acceptance of cyber attachments, such as active exoskeletons, is low. However, Eric Burns and his colleagues clearly demonstrated visual perception's dominance over proprioceptive cues during manual operations in virtual environments.[16] Representing collisions with virtual objects only visually thus might be sufficient in many applications, but this approach is likely to slow down user performance.

Graspable Input Devices

Although data gloves can be understood as the most direct way of interacting with virtual objects, handheld input devices are often a better alternative. Such devices don't need to be mounted or calibrated to individual user physiognomies. This reduces setup time and lowers the threshold for user involvement. Furthermore, users can rapidly pick them up, exchange them, and stow them when they're no longer needed. The way an input device is grasped also provides tactile information to the operating hands, thus guiding the interaction if the device is designed appropriately.

Whereas direct manipulation of virtual objects

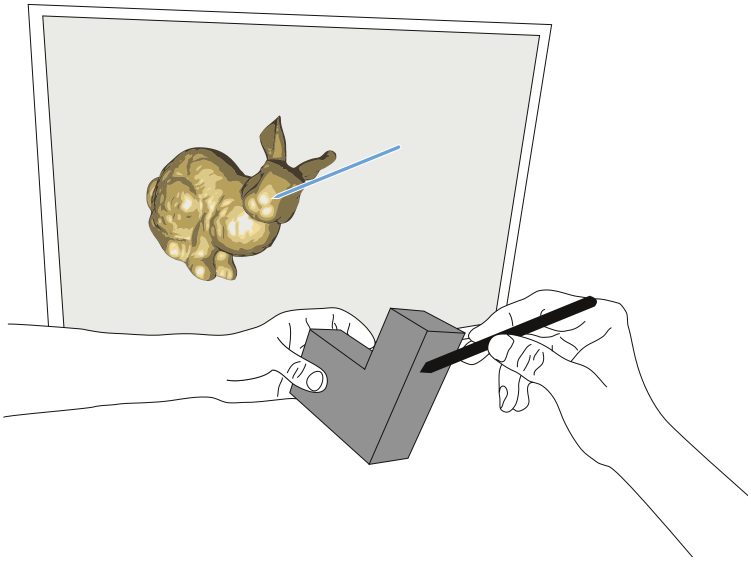

Figure 6. Object props can improve 3D object manipulation significantly because they provide a tangible reference to 3D computer graphics. Rough geometric approximations of the corresponding virtual object are mostly sufficient and even preferable for practical reasons.

with the hands features body awareness and skills as well as environmental awareness and skills, graspable input devices also involve the RBI theme of naive physics in terms of tangible physical properties. Such devices fall into three classes: object props, tool props, and digital foam.

Object props. These devices are physical artifacts resembling an object in the virtual world. For example, a user can change a virtual object's position and orientation by manipulating a tracked physical object of a similar shape (see Figure 6). A high level of similarity helps the user recognize tactile references relating to the virtual object's details, thus supporting fine-grained operations. But producing highly accurate real-world representations of virtual objects is often difficult. Rough geometrical approximations of the virtual object's shape are often more practical and potentially even preferable. Abstract or generic shapes let an input device represent various virtual objects.

Tool props. These devices don't refer directly to an application's content but to tools for that content's manipulation. In the real world, we employ many physical tools for specialized operations.

For computer applications, we can reduce the number of tools while maintaining comparable versatility by exploiting the fact that tools have two aspects: the handle and the effector. Because more variation exists among effectors than among handles that are adequate for certain tasks, we can use the same handle with different effectors. The mouse is a good example of this concept of toggling between modes for pointing, drawing, or other operations based on motion input.

The wand is a generic handle analogous to the mouse but for interaction with immersive 3D computer applications. In general, the wand position and orientation are tracked for controlling a virtual ray emanating from the wand or moving virtual objects in a 3D space.[3] Generic handles are useful for many applications, but the increasing popularity of more specialized interaction tools in 2D and 3D UIs also indicates users' desire to incorporate more sophisticated manual skills when interacting with computers.

A handle's physical shape affords a certain usage. An input device's design should therefore match its intended task's requirements. From the design of tools for physical action, we can draw out many principles for designing 3D computer input devices. For instance, small sizes afford precision grasping with the fingertips and thus foster fine motor skills. Spherical shapes imply rotations, whereas longer shapes define a direction—for example, for pointing. However, unlike many real-world tasks, computing applications usually don't require strong manual forces. Instead, the design of handheld 3D input devices must aim to minimize the tremor and fatigue common to nonsupported manual operations in mid air. The design space of adequate shapes for 3D input devices is still relatively unexplored.

One promising research direction for exploiting tangible input devices' physical properties is to toggle interaction modes on the basis of the device's spatial orientation and performed gestures.[17,18] You could also assign such mode changes through explicit switching, but basing them on how the user holds and operates the device implicitly provides user awareness. Instead of forcing the user to mentally keep track of frequently adjusted system states, this approach exploits context information that the user already has from passive haptic feedback of the input device.

Push buttons and sensors such as sliders, joysticks, and touch sensors can extend tool prop (and object prop) functionality.[3] This approach is often sensible if a group of canonical interaction tasks requires the control of additional DOF that could otherwise be supported only by additional devices or frequent mode changes. Similar to real-world tools, such specialized input devices support specific tasks better, but their application is less generic.

Digital foam. Recently, researchers have proposed sensor devices that track deformation forces on their surface.[19,20] Typically, these digital-foam devices capture finger impressions and can serve as object props for shaping virtual clay. Grasp recognition, another application of these devices, combines tactile feedback with potentially high-resolution input.[19] Furthermore, the sensor data can be used to determine how people grasp and manipulate physical objects.[20] This extends the concept of changing tool functionalities on the basis of the user's handling of the device. Because different functionalities might require grasping the tool differently, it makes sense to exploit this information in interactive computer applications.

Imagination-Based Interaction

Not only in computer applications but equally in the physical world, the benefits of increased productivity and comfort often come at the price of decreased accountability. A weight lifted by a machine can't be directly estimated. Covering distance with a subway often results in the loss of comprehension of geographical relations. The actual interdependencies of many of today's facilities are largely hidden to the user. Computing is based on abstraction and automation, so we generally don't know what happens "under the hood" of an application we use. All confidence in its functioning and reliability can only be based on a simplified concept.

Most well-established concepts and metaphors in human-computer interaction emerged from office work (generally 2D paperwork on a desk) and are often inadequate for 3D applications. But comprehensible metaphors can have other sources. These include references to popular science-fiction stories, common knowledge about technological achievements, and the arbitrary combination of aspects from previously learned patterns. Daily use of electronic commodities has also established advanced functionalities such as remote control and automated processes that you can invoke by command. As I mentioned before, further technically augmented or "magical" techniques such as scaling are often desired in 3D graphics applications.

Many functionalities that can be assigned to virtual objects aren't applicable in physical reality. 3D graphics can be instantly recolored or turned invisible. Models can be replicated, divided, or distorted. Objects that aren't required anymore can vanish without a trace. Unrealistic actions aren't necessarily impossible. As long as they're computable and users can imagine or vaguely understand the resulting functionality, they might be useful. Being limited to reality-like interaction would thus be a shortcoming for most 3D UIs—or as Ben Shneiderman put it, "Why Not Make Interfaces Better Than 3D Reality?"[21]

From a nonprogrammer's viewpoint, computing provides a virtually unconstrained option space. Consequently, many computer-augmented techniques might be hard to understand because users can't relate them to previous experiences. A consistent higher-level logic might help users recognize corresponding interaction patterns and guide their assumptions about possible functionalities.

A classification of such "magic" techniques might eventually lead to interface design guidelines that promote higher-level consistency, and thus usability, for a wide range of interfaces. Complementary to RBI's four themes, I suggest the following themes of IBI: *suspension of naive physics*, *scaling* (of geometry and motion), *automation*, *magic spells*, and *mode changes*.

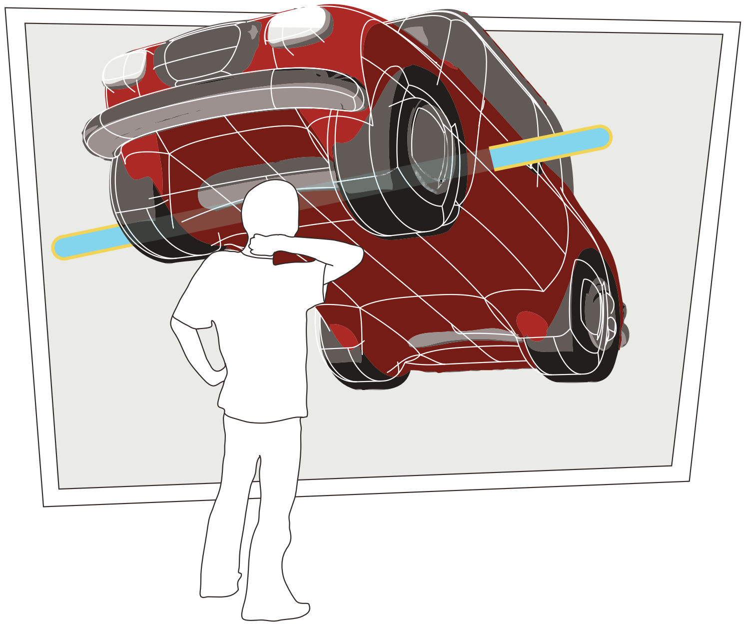

Figure 7. Virtual objects can be fixed on an arbitrary axis in mid air if that's the most suitable place for the task—in this case, examining a car's underbody.

Suspension of Naive Physics

Constraints such as simulated gravity and collision with geometry often facilitate interaction with 3D computer applications. However, the idea of having superpowers that let us walk through walls or become invisible is older than computing. In 3D computer applications, such old dreams and fairy tales have become a reality.

If not explicitly implemented, physical laws don't apply to virtual 3D environments. You can place objects wherever you want—even in mid air, if that's the most convenient place—and you can fly from one place to another to get a better overview. For example, you don't need a supporting structure to elevate a virtual car to examine its underbody (see Figure 7). Instead of sticking to gravity or collision, you can define constraints freely according to how they best support the user's task. CAD systems, for example, offer manipulation widgets that promote motion input on separate axes. This is similar to physical rails or bearings but takes far less effort to apply.

Geometric Scaling

Computer graphics can be scaled up to present more detail and improve accuracy. With the advent of multitouch interaction, this approach became popular in 2D graphics applications. We can describe the functionality of such zooming UIs in terms of scaling the graphical environment or adjusting the viewing distance to the virtual canvas.

In perspective 3D graphics, zooming, scaling, and changing the viewing distance are very different actions. Although all three affect the projected objects' size, they alter different elements of the 3D scene. Zooming adjusts the camera's field of view, scaling changes the objects' size, and through viewpoint navigation, users can adjust the distance to objects of interest.

In monoscopic 3D graphics, the resulting changes to the view might still appear equivalent. Unless viewpoint navigation provides additional depth cues through motion parallax, users can't differentiate whether they're looking at a toy car or a real-size car model because the images on the screen might be identical.

With stereoscopic viewing, the situation is clearly different. The eye distance in relation to the object size defines how big an object appears. You could say that eye distance is the unit of stereoscopic virtual worlds. Moving toward an object doesn't change its apparent size. Scaling an object makes it really larger, and a toy car might turn into a real car—true magic, if it weren't merely an image-based illusion.

Geometric scaling is popular in 3D computer graphics, but no particular interface concept for it has achieved broad acceptance yet. The many different approaches include arbitrary button mappings, mouse-operated widgets, and virtual system-control panels. A consistent logic behind these different attempts would ease the use of interfaces for scaling. A good starting point is the idea of stretching and squeezing as applied in multitouch 2D interfaces.

Motion Scaling

Without affecting the view, we can also tackle the trade-off between accuracy and rapidity by scaling only the motion input. The computer system can't directly influence the user's physical action, but we can scale down the resulting motion in screen space for high accuracy or accelerate it for rapidity. The control-display gain (CD gain) defines the relation of user input to displayed motion. It can be set explicitly or continuously adapted—implicit with the induced motion's current velocity.

The latter approach is based on the observation that people perform fine-grained, accurate movements slowly, whereas high velocities occur only during coarse ballistic movements. The broad user acceptance of pointer acceleration provided by common 2D UIs confirms its general applicability. However, maintaining an absolute mapping between user input and displayed motion is often relevant.

A good example is 3D pointing. Varying the CD gain during 3D pointing would result in diverging pointing directions of the displayed ray and the user's hand, thus eliminating proprioceptive cues. In 3D, objects often need to be manipulated without physical support for the hands operating in mid air. Hence, increasing accuracy through downscaled motion input can be particularly beneficial in that context. Scott Frees and G. Drew Kessler introduced PRISM (Precise and Rapid Interaction through Scaled Manipulation), a transfer function that decreases motion velocity only during slow, fine-grained adjustment phases and compensates for the resulting offset during rapid ballistic-motion phases.[22]

Automation

Automation already became part of our daily routine with the advent of machinery. Information processing only advanced this concept in that the same device can now perform a quasi-unlimited number of frequently changing actions, whereas mechanical machines must be built specifically for one purpose. This makes a huge difference for the UI. Whereas mechanical machines are predictable owing to a consistent identity, estimating the automated processes of computing is difficult.

In computing, all that's required for automation is an adequate description of the steps involved. For interactive 3D computer applications, this implies, for example, that every item involved can immediately be animated without requiring a driving engine. Consequently, distances in virtual environments can be covered automatically. Selecting a target location (for example, through pointing; see Figure 8) is sufficient to get there. Obviously, in virtual environments, not only can viewpoint motion be facilitated, but also any other transition of objects and parameters can be automated.

Automated motion can also employ rate control. Instead of only defining a target destination, the user can continuously control direction and velocity. Rate control enables positioning in three dimensions without forcing the user to operate input devices in mid air. So, the user's hands can be physically supported (for example, on a desk) for higher comfort. Shumin Zhai demonstrated that rate control works best if the input device provides feedback in terms of a counterforce to the induced motion.[23]

Magic Spells

Computing allows dynamically assigning meaning and automated processing to any involved object and thus adapting the application to the requirements of a variety of tasks. This major advantage over real-world interaction also promotes a different workflow. Instead of operating several mechanical tools in an orderly fashion, users are concerned more with assigning parameters to selected objects.

Having selected a target item by pointing at it, users can apply commands. If the target item or location (for example, for automated travel or to change an object's color) isn't visible from a current viewpoint, the user can also use abstract representations of 3D objects and environments to select it. These representations include maps, worlds in miniature, and simple lists of items.[3]

Combined with speech input,[24] such selection-and-command interfaces are very effective. They empower users to operate just as magicians cast magic spells, by pointing with a wand at target objects or their symbolic representations.

More common and still very effective is selecting commands from menu structures. This implies handling the task in two subsequent steps, whereas pointing and speaking can be performed in parallel. Nevertheless, command selection from graphical menus has practical benefits. Besides being computationally less expensive than speech recognition, visualizing the option space relieves users from memorizing numerous commands.

Mode Changes

Because the properties of objects and tools in computer applications might frequently change, accountability for such changes becomes a major issue. For example, consider the Homer (*H*and-Centered *O*bject *M*anipulation *E*xtending *R*ay-Casting) technique, which combines the advantages of ray-based selection and hand-centered object manipulation.[3] After the user selects an object by pointing at it, the motion input of the user's hand is no longer applied to the virtual ray. Instead, it directly manipulates the selected object. The technique facilitates the handling of distant objects but breaks with the consistency of ray-based interaction.

For automated viewpoint navigation, Bowman and his colleagues have shown that an animated transition from one location to another increases the user's orientation in a virtual environment.[3] In contrast, sudden viewpoint changes might cause disorientation. Similarly, we can assume that an explicit presentation of transitions from one behavior to another can support the user's understanding of changing system states. For example, in Homer, a virtual-hand representation might move along the ray toward the selected object. More important than emphasizing the transition between system states is representing the respective interaction modes appropriately. For experts working with a known set of interface states, it's mostly sufficient if those states are easily distinguishable. But if the extent of fluently adapting functionalities and properties becomes large, or the user has no experience with the system, the representations should also be meaningful outside the application context. Furthermore, emphasizing the transitions of system states can help support the user's cognitive processes.

The optimal length and emphasis for presenting such state transitions might depend on several dynamic factors such as the user's expertise, concentration, or mood. However, researchers haven't thoroughly studied how state transitions affect user performance or how to dynamically predict good parameters for presenting these transitions. At the very least, changes of system states should follow a comprehensible logic. In the case of Homer, the logic involves the idea of moving the hand to the object that was just selected. In the best case, the changes are even predictable.

The Bubble technique, for example, combines position control for short-range manipulations with rate control to cover larger distances with input from a force feedback device.[25] A spherical virtual workspace surrounds the manipulated 3D cursor. Within that sphere, the user applies position control. Pushing the cursor against the borders results

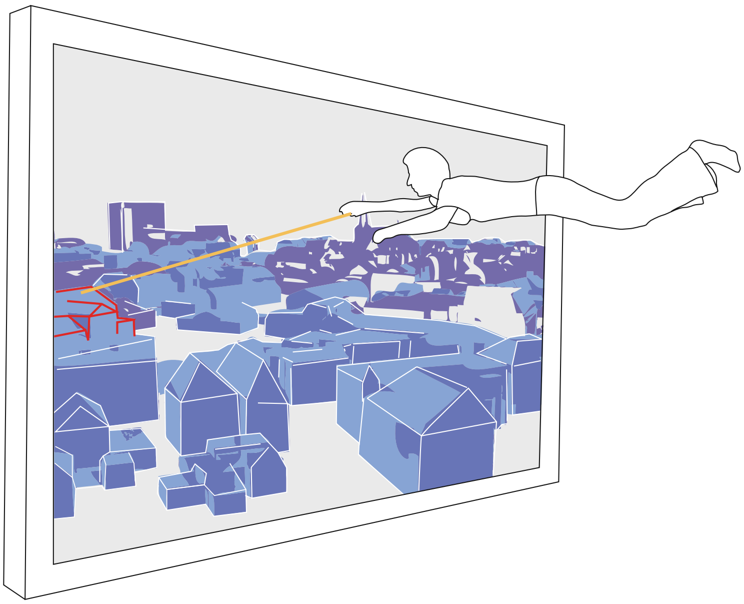

Figure 8. Computing enables "flying" automatically to a selected destination.

in a counterforce applied by the device and rate-controlled motion of the virtual workspace in the induced direction. This hybrid interface's comprehensible logic, combined with the visualization of the sphere defining the range of applied position control, results in highly predictable behavior.

In designing 3D UIs that combine the advantages of RBI and IBI, the challenge is to mediate between model consistency and functional expressiveness. Future research on navigation in virtual environments should therefore also consider the users' traveling between different mental models—imaginary worlds with varying constraints and possibilities.

Also, the physical I/O devices constitute the tangible reality of human-computer interaction. UIs therefore can't be designed only in screen space. Besides input devices, the technology and design of display devices also significantly affect an application's usability. In particular, the development and availability of stereoscopic multiviewer projection technology is a major step toward truly collaborative colocated systems.[26] Such systems facilitate social and spatial interaction because they better build on environmental awareness and skills as well as social awareness and skills. This new, largely unexplored design space of 3D UIs for colocated interaction includes exciting topics such as mutual awareness and accountability of actions, private information spaces in shared environments, and group navigation techniques.

Acknowledgments

This article wouldn't have been possible without the strong support of my friends and colleagues in the 3D UI research community. I particularly thank Regis Kopper for his comments on an early draft and Daniel Fischer for discussions about the different viewpoints on human-computer interaction in general. Bernd Froehlich, Doug Bowman, and the anonymous reviewers made many comments that helped me improve the article's structure and emphasize the most relevant points.

References

1. B. Shneiderman, "Direct Manipulation: A Step beyond Programming Languages," *IEEE Computer*, vol. 16, no. 8, 1983, pp. 57–69.
2. A. Kulik et al., "The Influence of Input Device Characteristics on Spatial Perception in Desktop-Based 3D Applications," *Proc. 2009 Symp. 3D User Interfaces* (3DUI 09), IEEE Press, 2009, pp. 59–66.
3. D. Bowman et al., *3D User Interfaces: Theory and Practice*, Addison-Wesley, 2005.
4. R.J.K. Jacob et al., "Reality-Based Interaction: A Framework for Post-WIMP Interfaces," *Proc. 26th Ann. SIGCHI Conf. Human Factors in Computing Systems* (CHI 08), ACM Press, 2008, pp. 201–210.
5. M. Hachet et al., "Navidget for 3D Interaction: Camera Positioning and Further Uses," *Int'l J. Human-Computer Studies*, vol. 67, no. 3, 2009, pp. 225–236.
6. R. Kopper et al., "Design and Evaluation of Navigation Techniques for Multiscale Virtual Environments," *Proc. 2006 IEEE Conf. Virtual Reality* (VR 06), IEEE CS Press, 2006, pp. 175–182.
7. A. Kunert et al., "A Comparison of Tracking- and Controller-Based Input for Complex Bimanual Interaction in Virtual Environments," *Proc. EG-ACM Symp. Virtual Environments* (IPTEGVE 07), Eurographics Assoc., 2007, p. 4352.
8. A. Kunert et al., "Facilitating System Control in Ray-Based Interaction Tasks," to be published in *Proc. 2009 ACM Symp. Virtual Reality Software and Technology* (VRST 09), ACM Press, 2009.
9. I.E. Sutherland, "A Head-Mounted Three Dimensional Display," *Proc. Dec. 9–11, 1968, Fall Joint Computer Conf., Part I*, ACM Press, 1968, pp. 757–764.
10. J. Yoon and J. Ryu, "A Novel Locomotion Interface with Two 6-DOF Parallel Manipulators That Allows Human Walking on Various Virtual Terrains," *Int'l J. Robotics Research*, vol. 25, no. 7, 2006, pp. 689–708.
11. M. Slater, A. Steed, and M. Usoh, "The Virtual Treadmill: A Naturalistic Metaphor for Navigation in Immersive Virtual Environments," *Selected Papers of the Eurographics Workshops on Virtual Environments '95*, Springer, 1995, pp. 135–148.
12. M. Usoh et al., "Walking > Walking-in-Place > Flying, in Virtual Environments," *Proc. Siggraph*, ACM Press, 1999, pp. 359–364.
13. S. Razzaque et al., "Redirected Walking in Place," *Proc. 2002 Workshop Virtual Environments*, Eurographics Assoc., 2002, pp. 123–130.
14. G. Bruder, F. Steinicke, and K.H. Hinrichs, "Arch-Explore: A Natural User Interface for Immersive Architectural Walkthroughs," *Proc. 2009 IEEE Symp. 3D User Interfaces* (3DUI 09), IEEE Press, 2009, pp. 75–82.
15. S.M. Peck, C. North, and D. Bowman, "A Multiscale Interaction Technique for Large, High-Resolution Displays," *Proc. 2009 Symp. 3D User Interfaces* (3DUI 09), IEEE Press, 2009, pp. 31–38.
16. E. Burns et al., "The Hand Is Slower Than the Eye: A Quantitative Exploration of Visual Dominance over Proprioception," *Proc. 2005 IEEE Conf. Virtual Reality* (VR 05), IEEE CS Press, 2005, pp. 3–10.
17. X. Cao and R. Balakrishnan, "VisionWand: Interac-

tion Techniques for Large Displays Using a Passive Wand Tracked in 3D," *Proc. 16th Ann. ACM Symp. User Interface Software and Technology* (UIST 03), ACM Press, pp. 173–182.
18. J. Rekimoto and E. Sciammarella, "ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices," *Proc. 13th Ann. ACM Symp. User Interface Software and Technology* (UIST 00), ACM Press, 2000, pp. 109–117.
19. D.K. Pai et al., "The Tango: A Tangible Tangoreceptive Whole-Hand Human Interface," *Proc. 1st Joint Eurohaptics Conf. and Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems* (World Haptics 05), IEEE Press, 2005, pp. 141–147.
20. B.T. Taylor and V.M. Bove Jr., "Graspables: Grasp-Recognition as a User Interface," *Proc. 27th Int'l Conf. Human Factors in Computing Systems* (CHI 09), ACM Press, 2009, pp. 917–926.
21. B. Shneiderman, "Why Not Make Interfaces Better Than 3D Reality?" *IEEE Computer Graphics and Applications*, vol. 23, no. 6, 2003, pp. 12–15.
22. S. Frees and G.D. Kessler, "Precise and Rapid Interaction through Scaled Manipulation," *Proc. 2005 IEEE Conf. Virtual Reality* (VR 05), IEEE CS Press, 2005, pp. 99–106.
23. S. Zhai, "User Performance in Relation to 3D Input Device Design," *ACM Siggraph Computer Graphics*, vol. 32, no. 4, 1998, pp. 50–54.
24. R.A. Bolt, "'Put-That-There': Voice and Gesture at the Graphics Interface," *Proc. Siggraph*, ACM Press, 1980, pp. 262–270.
25. L. Dominjon et al., "The 'Bubble' Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace," *Proc. 1st Joint Eurohaptics Conf. and Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems* (WCH 05), IEEE CS Press, 2005, pp. 639–640.
26. B. Fröhlich et al., "Implementing Multi-viewer Stereo Displays," *Proc. 13th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision* (WSCG 05), Union Agency Science Press, 2005, pp. 139–146.

Alexander Kulik *is a PhD student in Bauhaus-Universität Weimar's Virtual Reality Systems Group. His research interests are user behavior in virtual environments, users' cognitive models of tasks, and the design of appropriate interaction tools. He received his Diploma in industrial design from Bauhaus-Universität Weimar. He's a member of the ACM. Contact him at kulik@uni-weimar.de.*


Top articles, podcasts, and more.
cn
computingnow.computer.org