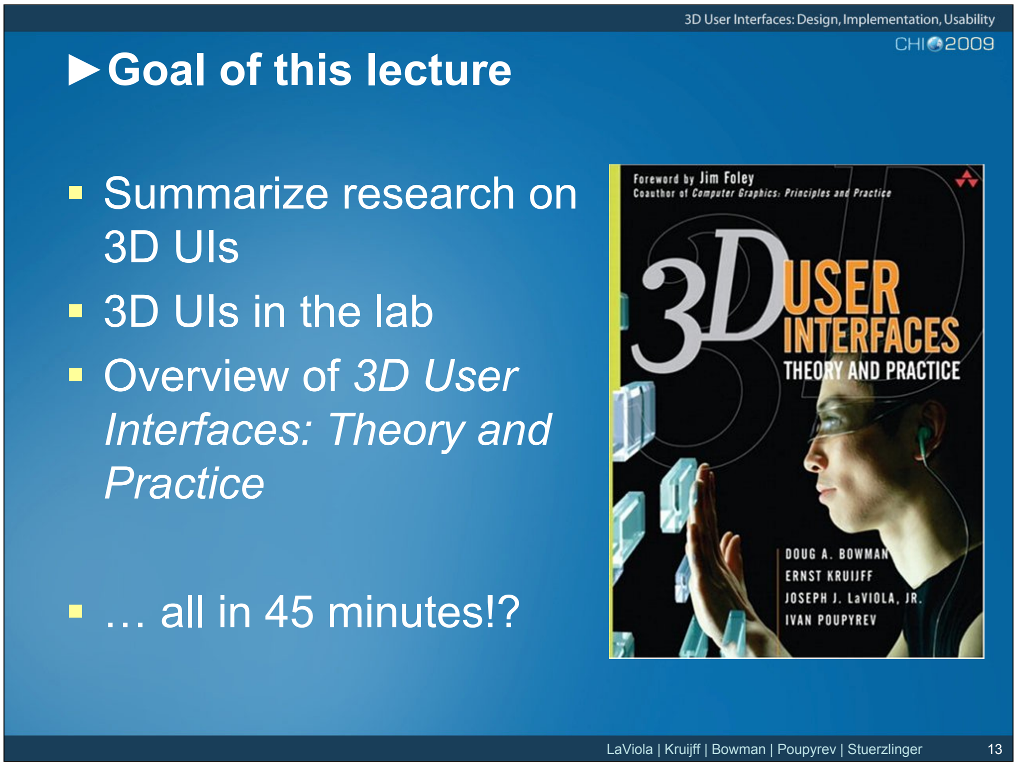
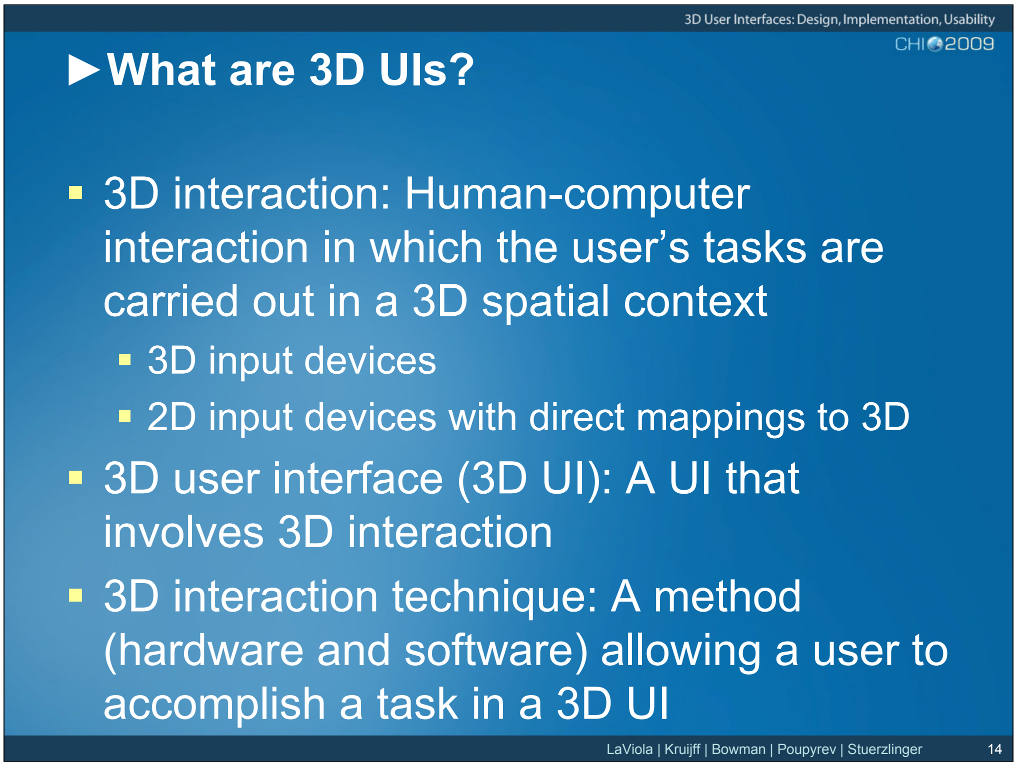
3D UIs 101

Doug Bowman

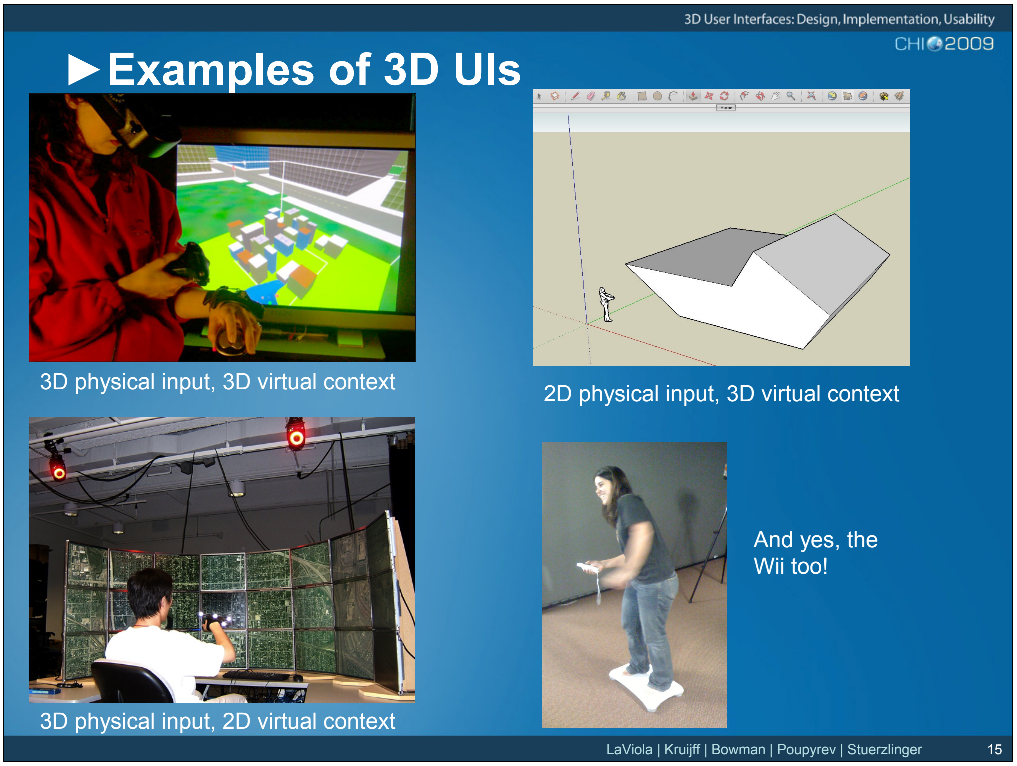
Welcome, Introduction, & Roadmap
3D UIs 101
3D UIs 201
User Studies and 3D UIs
Guidelines for Developing 3D UIs
Video Games: 3D UIs for the Masses
The Wii Remote and You
3D UI and the Physical Environment
Beyond Visual: Shape, Haptics and Actuation in 3D UI
Conclusion



The goal of this lecture is to provide a foundation for the rest of the course. It will provide a whirlwind overview of research on 3D UIs to date, using our book 3D User Interfaces: Theory and Practice as a guide. Given the limited time, we'll just present a few highlights, so that those not familiar with 3D UIs can understand the topics and issues presented in the rest of the course.



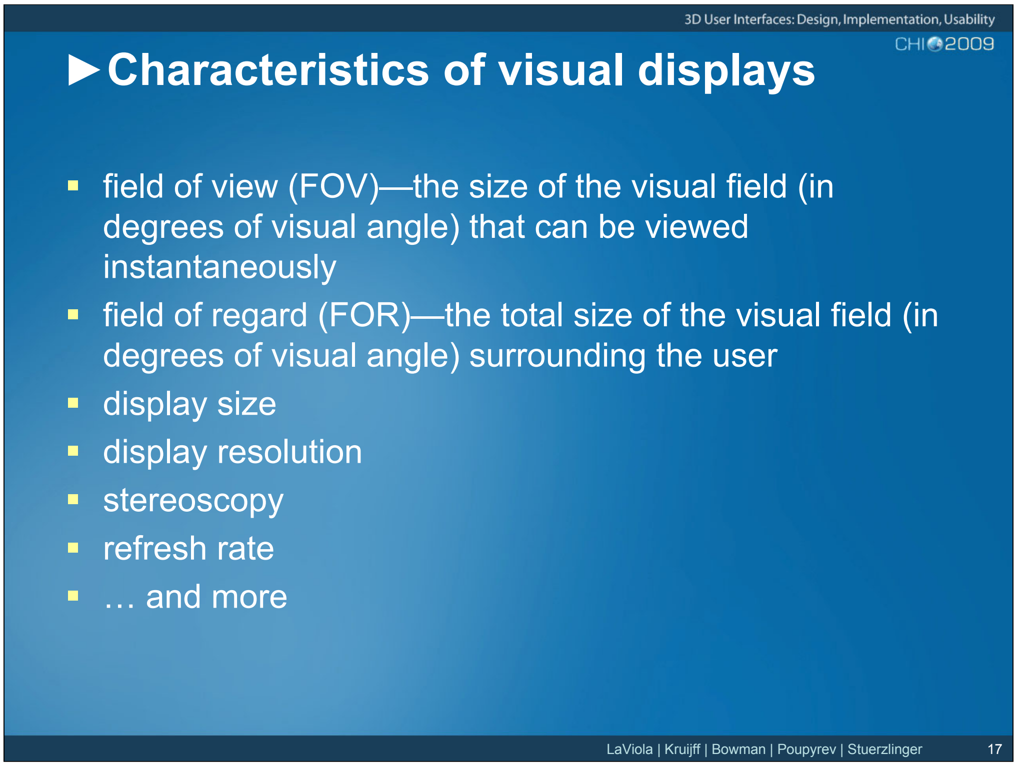
Our definitions of 3D UI and related terms.



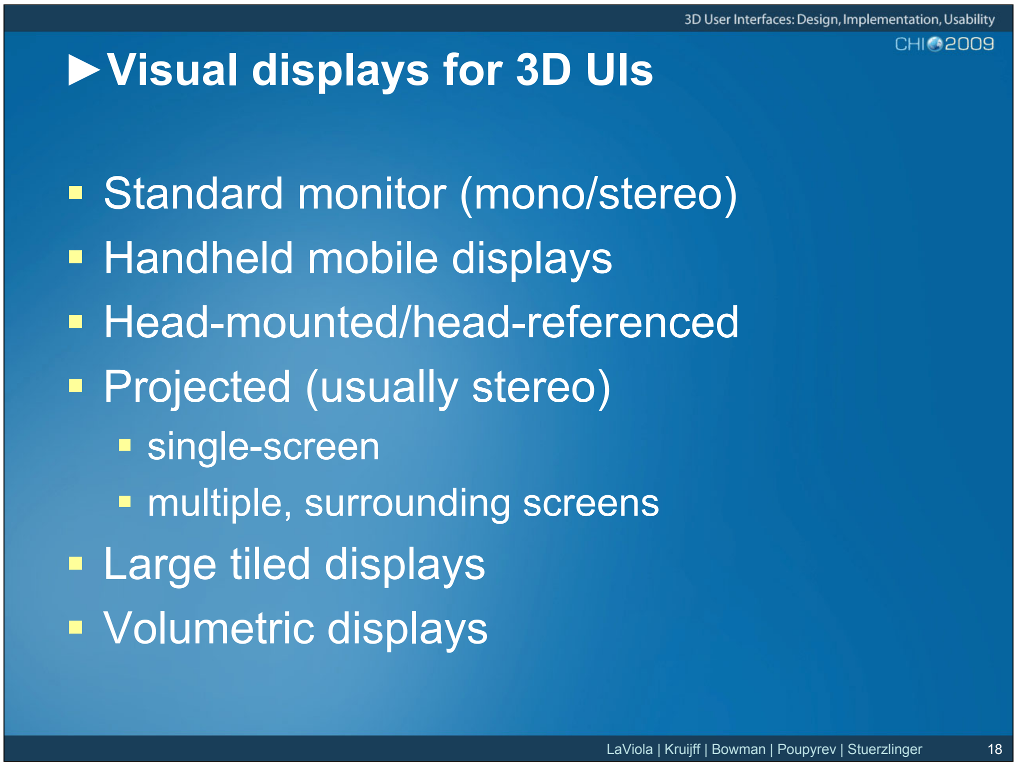
The definitions on the previous slide lead to three categories of user interfaces that we consider 3D UIs:

1. 3D input devices are used to interact with a 3D virtual world
2. 3D input devices are used to interact with a 2D virtual world
3. 2D input devices are used to interact (directly) with a 3D virtual world

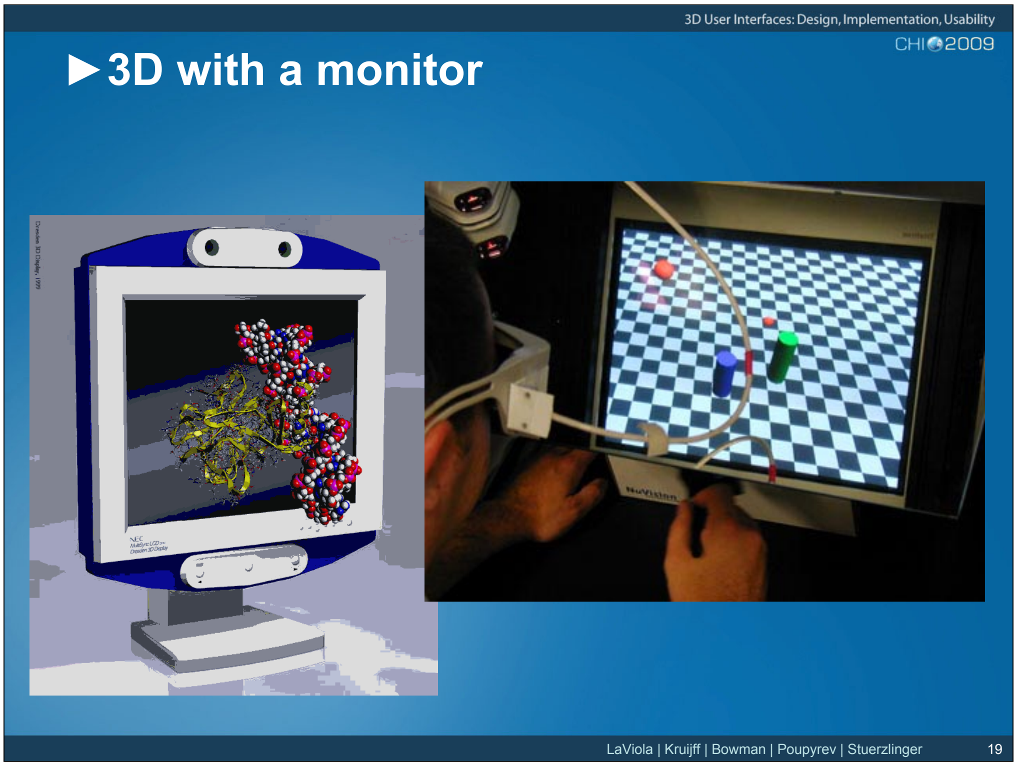

3D User Interfaces: Design, Implementation, Usability
CHI 2009
Display devices for 3D UIs
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
16



Although this is not an exhaustive list, it gives a sense of the ways that visual displays for 3D UIs can be characterized. It also provides a more or less standardized way to compare visual displays that are very different.

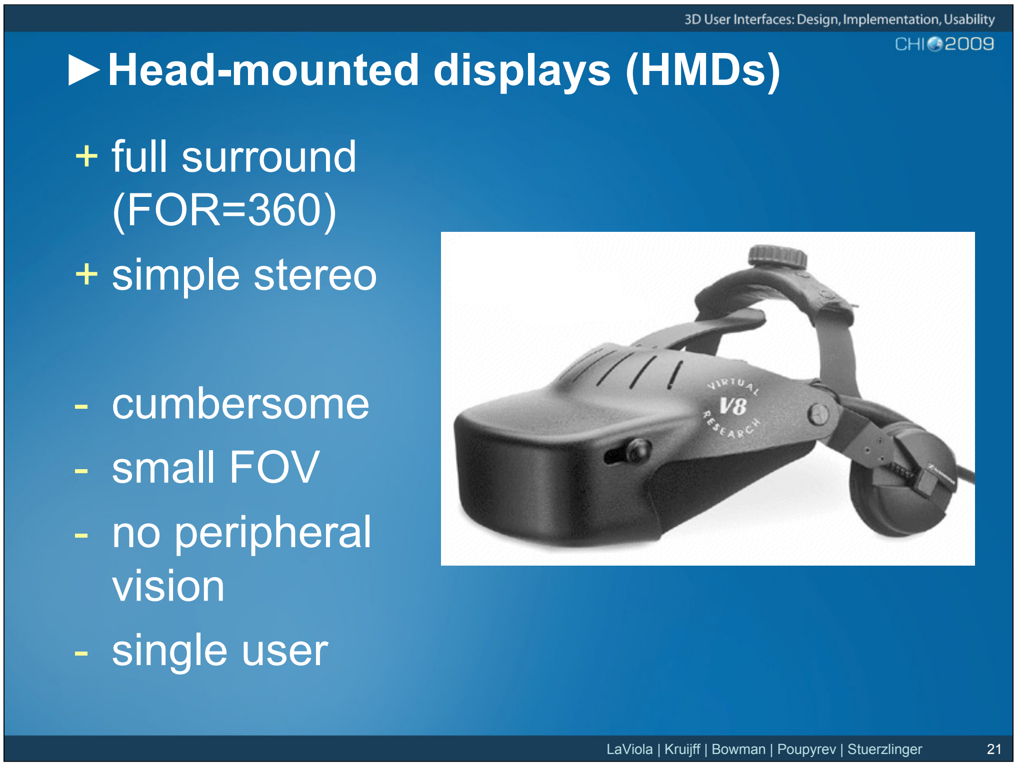


We'll summarize the pros and cons of a few of the more common and/or interesting visual displays for 3D UIs.



3D UIs on the desktop are easier to achieve now than ever before. There are commercially-available autostereoscopic displays, making 3D viewing without glasses feasible. Adding a head tracker produces so-called “fishtank VR,” and a handheld tracking device (such as the Wii Remote) allows 3D input as well.

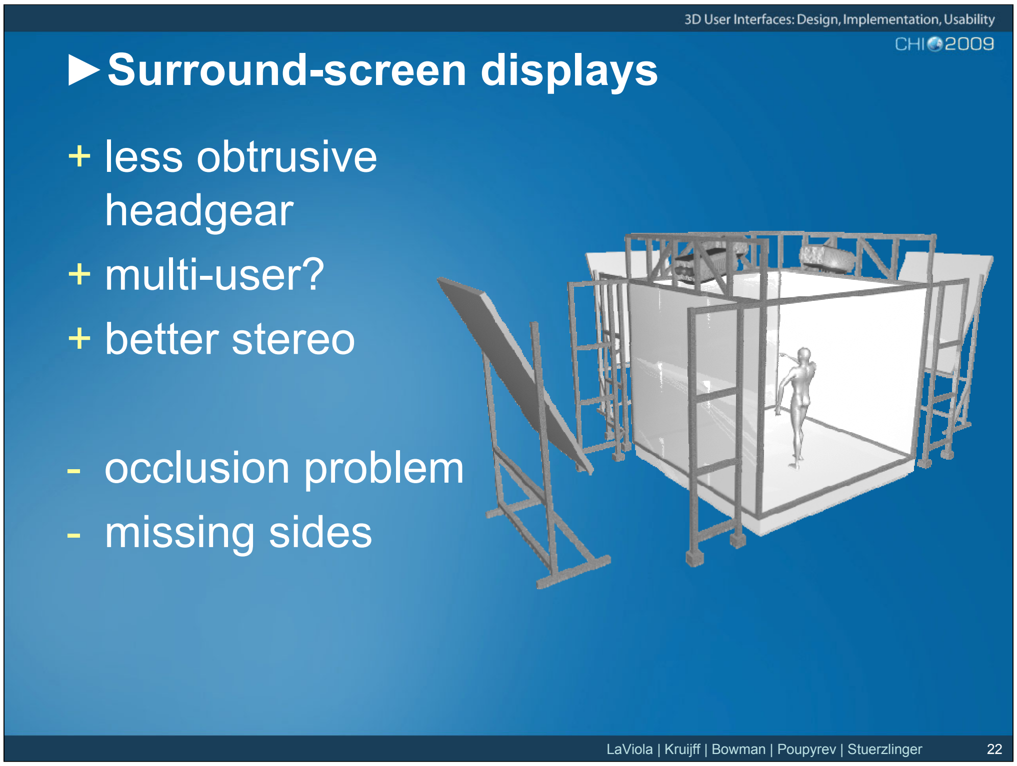
Powerful 3D graphics and 3D motion input (via accelerometers) or 3D position tracking (via vision-based trackers) are also available on handheld platforms like the iPhone, opening up a new realm of 3D UI possibilities.



One of the most common display devices used for 3D UI applications is the head mounted display (HMD). With a tracking device attached to the device, it produces a stereoscopic view that moves relative to the user's head position and orientation. Although traditionally the user cannot naturally see the real world, cameras are sometimes mounted on the HMD which allows it to display both real world video and graphical objects. In addition, some HMDs offer see-through options. This type of technology is used in augmented reality systems.

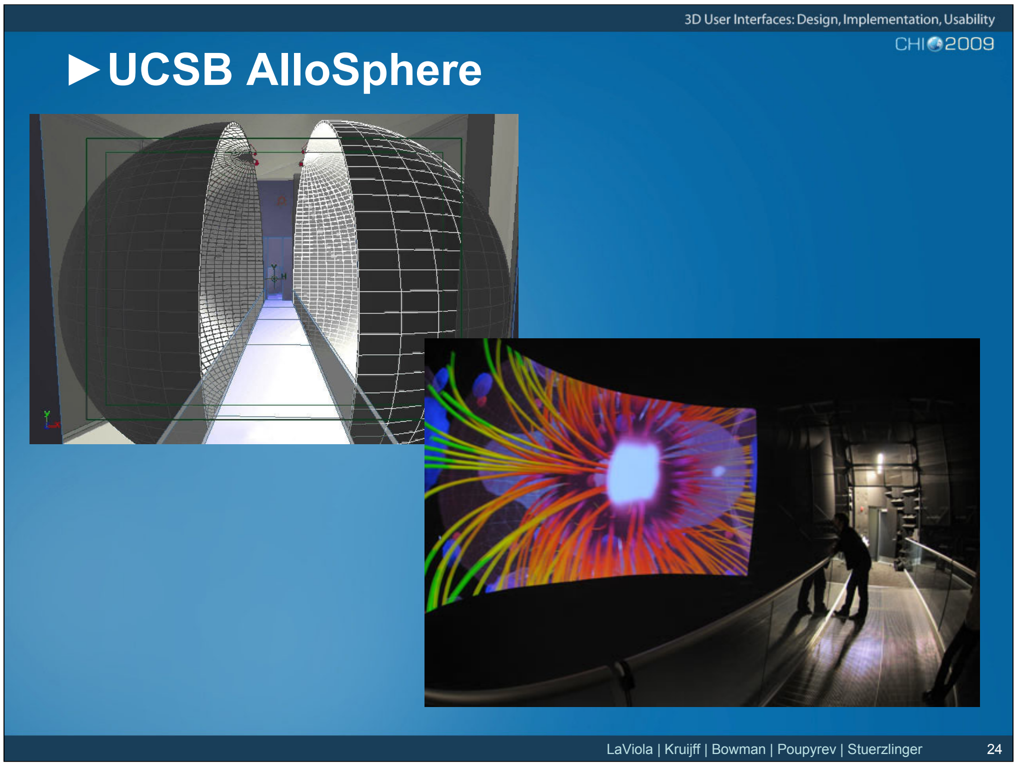
Since each eye is presented with one screen, HMDs allow for good stereoscopic viewing. These two screens are very close to the user's eyes (1 to 2 inches). As a result, all viewable objects are behind the screen so any object clipping will appear to the user as being outside his/her field of view. A big disadvantage of HMDs is that can get heavy very quickly and, unfortunately, the higher the HMD's quality, the heavier it usually is. Although HMDs are still used in many VR labs and entertainment centers, researchers and practitioners are rapidly moving towards projection-based display devices especially when high-resolution graphics are required.

Recently a high-resolution and wide FOV HMD came onto the market (www.sensics.com). It remains to be seen whether this will cause some high-end applications to return to HMDs.

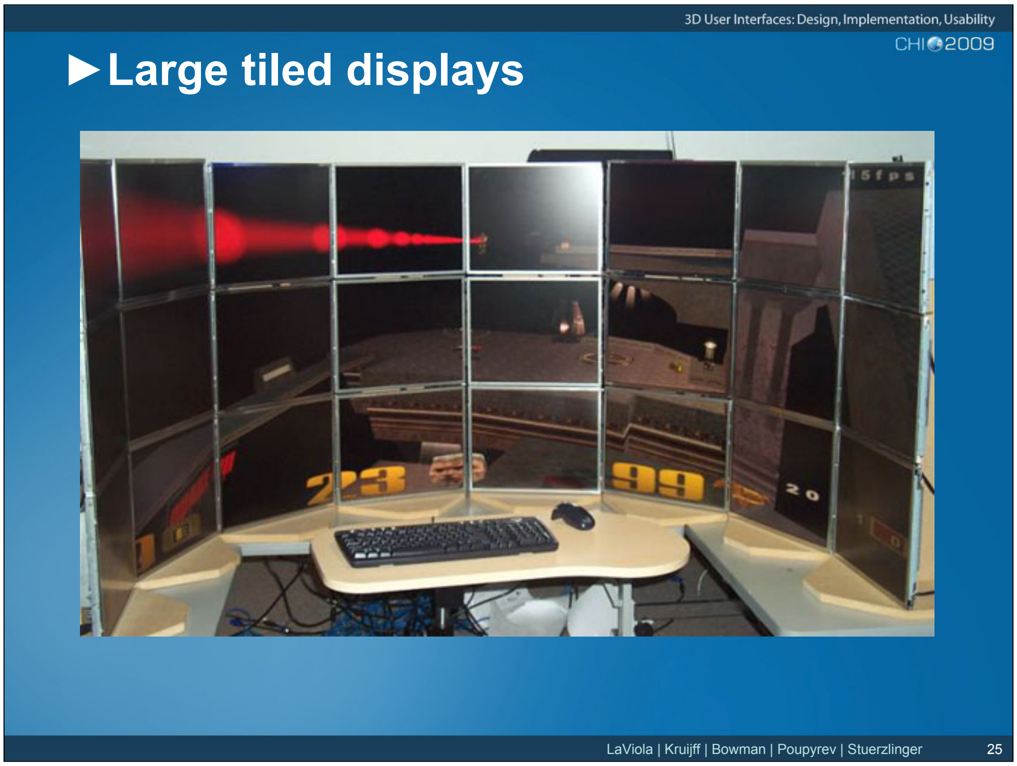


Surround-screen displays, such as the CAVE™ are also extremely popular. Instead of attaching the displays to the user, they place the displays in the world. Such displays are typically rear-projected, stereoscopic, and head tracked. They range from two-screen L-shaped configurations to semi-cylindrical displays to spherical displays.

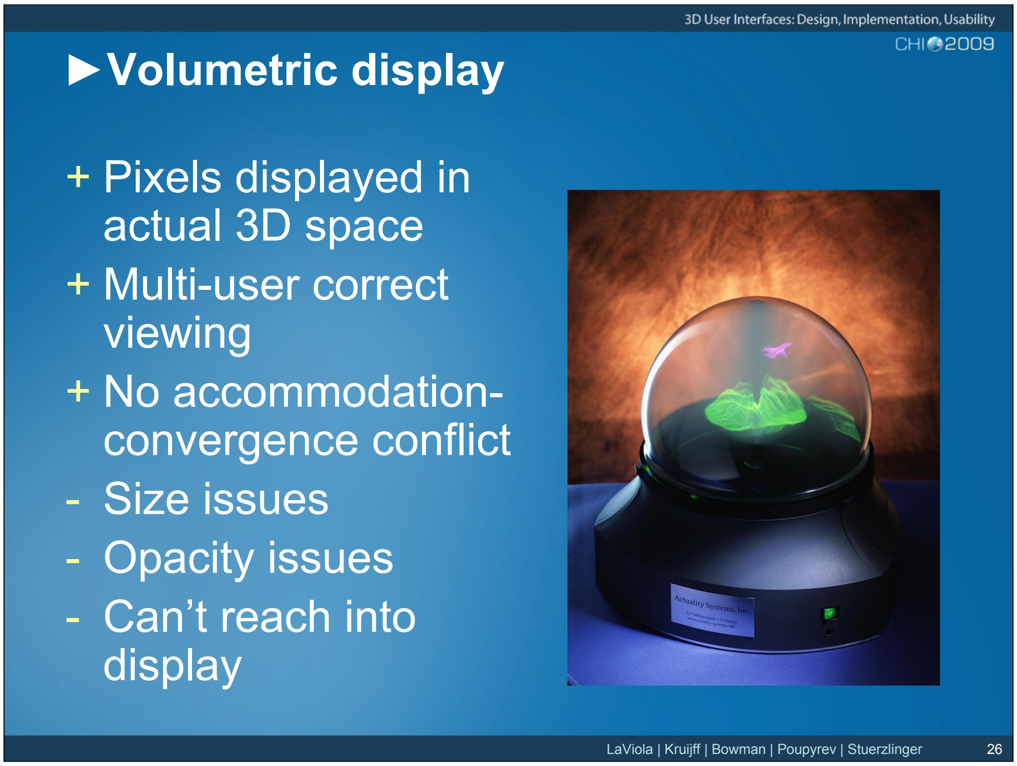
Traditionally, HMDs have one big advantage over surround-screen displays - a 360-degree field of regard (i.e., the graphics appear around the user in every direction). But this advantage was eliminated with the advent of fully-surrounding surround-screen displays, such as the six-sided DiVE at Duke University.



Another fully-surrounding display is the AlloSphere at UCSB. It's a 3-story high spherical display with a "bridge" running through the center. When it is completed, it will offer 360-degree surround with high-resolution audio and stereoscopic video.

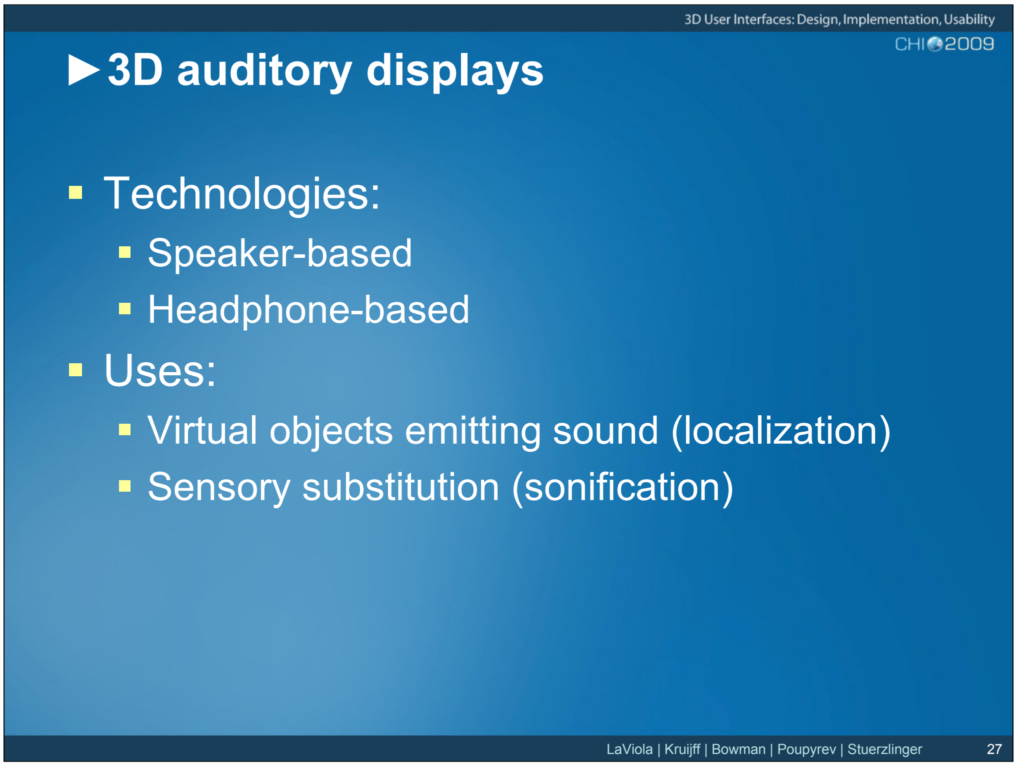


The cheapest way to get a large display with very high-resolution is to tile multiple panels together. Here, 24 LCDs (without their casings) are tiled to produce a large, curved “desktop” display with more than 46 million pixels. 3D applications can run on such displays with the help of a small cluster of PCs and software (e.g., Chromium) that distributes the graphics rendering to each machine.



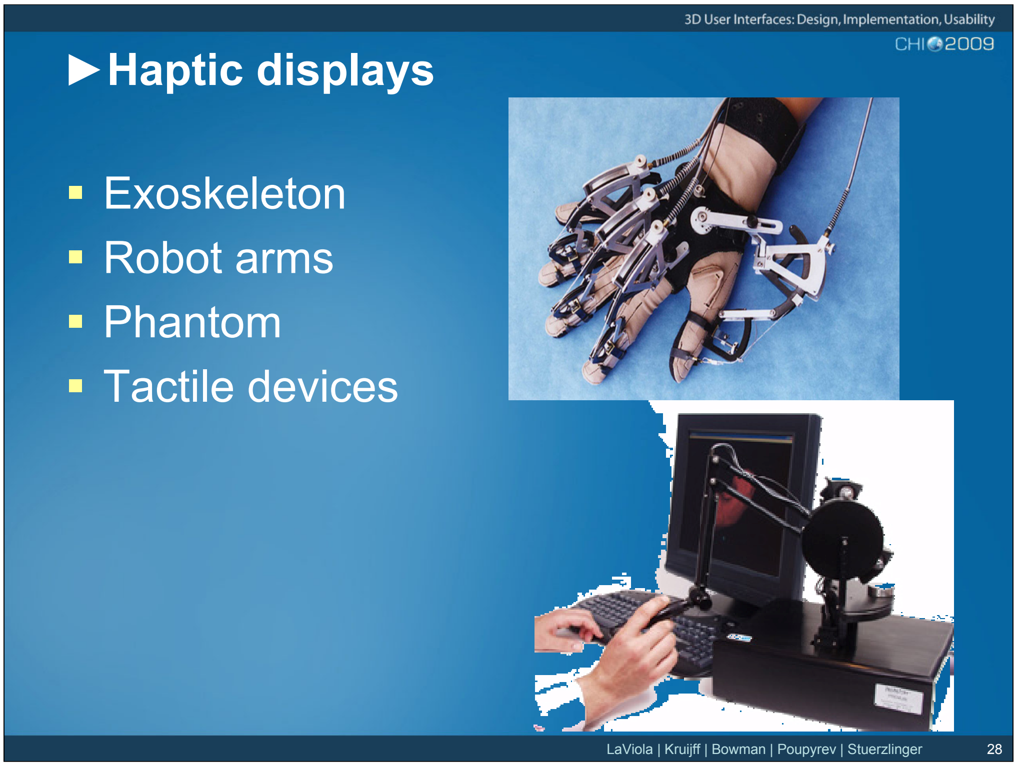
Volumetric displays produce a "truly 3D" image by actually illuminating locations in physical 3D space. The display shown here, from Actuality Systems, uses a rotating transparent display enclosed in a glass dome.

These displays solve a problem common to all other 3D display types - the accommodation-convergence mismatch. Accommodation is an oculomotor depth cue based on the depth of focus of the eye, while convergence, also an oculomotor cue, is based on the rotation of the eyes to look at a single object. In 3D displays that project stereoscopic images on a flat screen, accommodation and convergence are always in conflict (unless the object is at the depth of the screen). Volumetric displays provide correct accommodation and convergence cues.



There are a number of different ways in which a 3D auditory system can be set up. A simple setup is to use stereo head phones. However, this restricts usage to only one person at a time. Another setup is to place speakers in certain logistic areas around the environment. This setup allows for more than one user to take part in the experience but is somewhat more complicated to setup and write software for.

There are two different ways, localization and sonification, in which sound can be used as an output medium in virtual environment applications. In localization, the goal is to generate three dimensional sound. In sonification, the goal is to turn certain types of information into sounds.



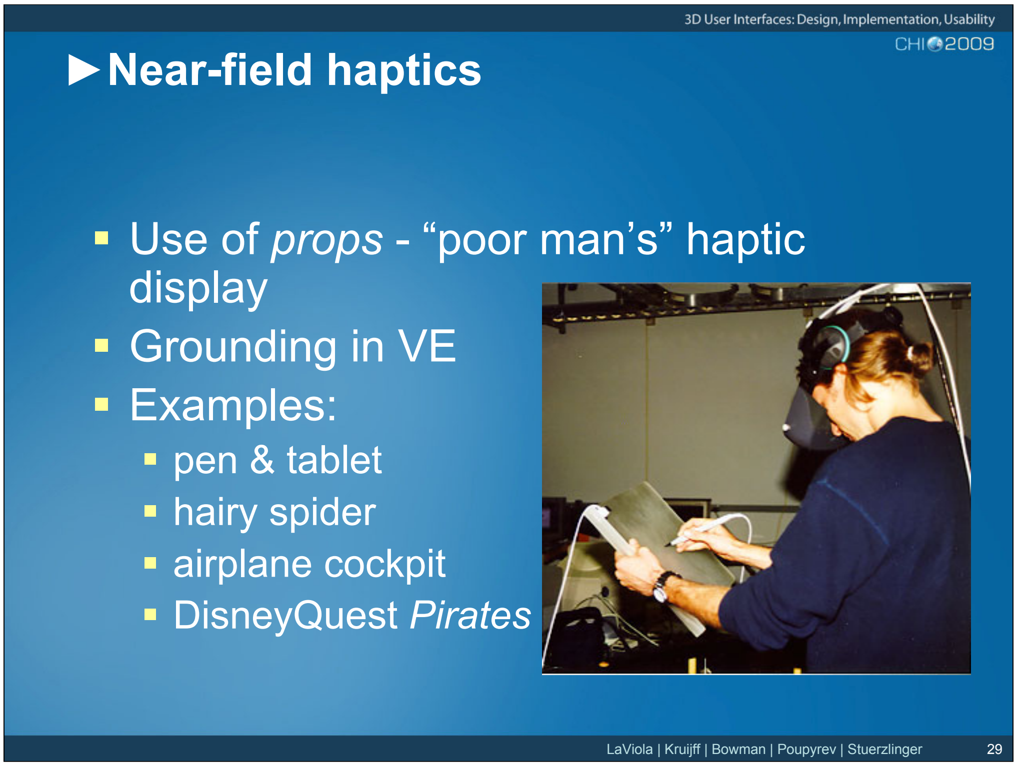
Haptics represents a critical component in virtual environment interaction. Allowing a user to touch and feel in the virtual world in the same way that they do in the physical world is extremely powerful. Unfortunately, haptic and tactile output device research has not made rapid progress.

There are essentially four different methods in which haptic and tactile feedback is generated. The first method is ground-referenced feedback which creates a physical link between the user and ground with the feedback relative to a single contact point. An example is the Sensable Phantom. The second method is body-referenced feedback which places a device on some part of the user's body. An example of a body-referenced haptic device is Virtual Technologies' CyberGrasp which is shown in the top picture. The third method for generating feedback is tactile which uses some type of oscillatory or vibrating device to stimulate the user's tactile sense. Finally, the last method of generating feedback is via dermal tactile which stimulates the user's nerves in the fingertips.

References:

www.sensable.com

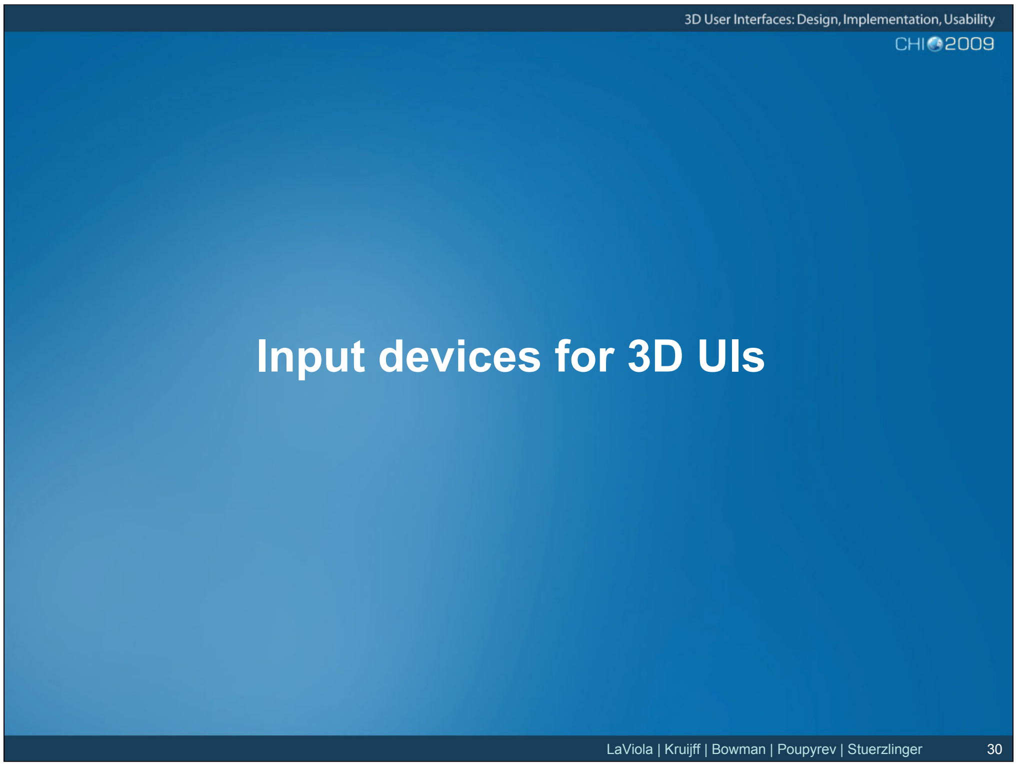
www.immersion.com

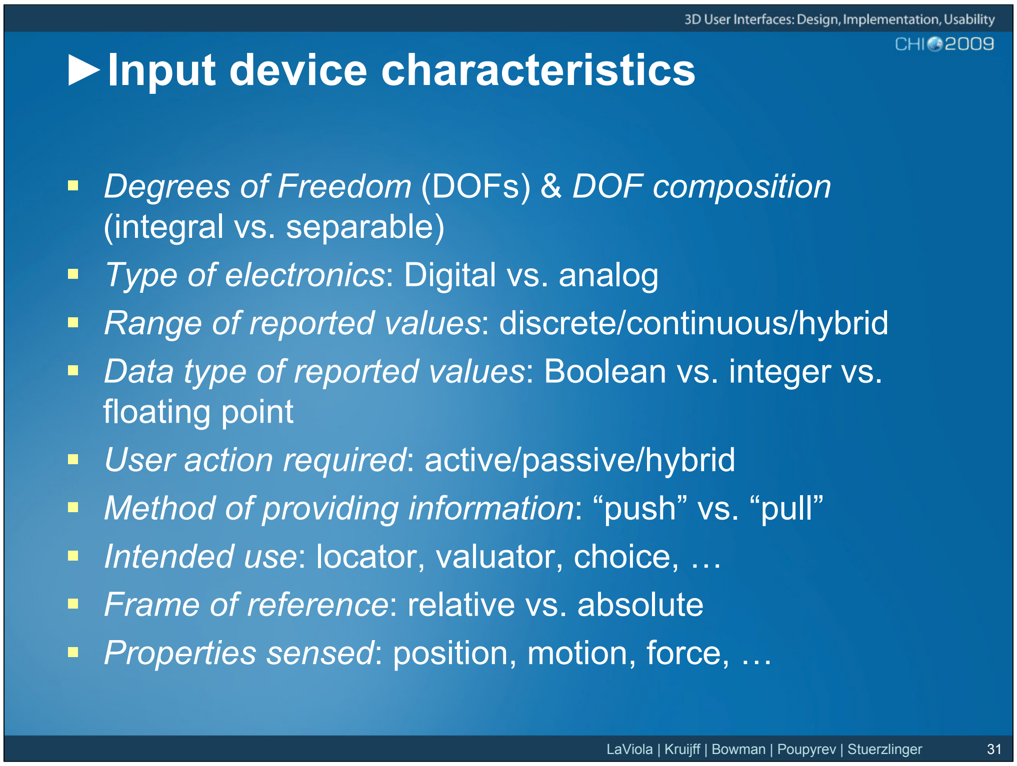


A simpler way to provide haptic feedback is the use of props - physical objects that represent their virtual counterparts. This is also called "near-field haptics" or "passive haptics." This has been an extremely important idea historically in 3D UIs.

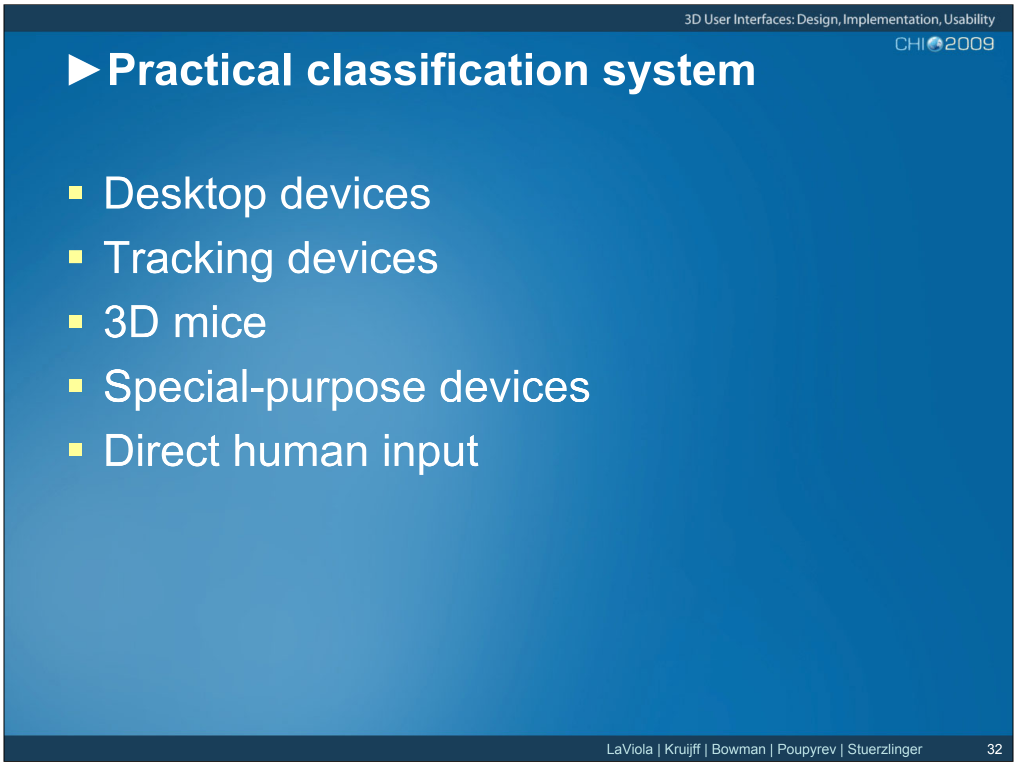
Hinckley, K., Pausch, R., Goble, J. and Kassell, N., Passive Real-World Interface Props for Neurosurgical Visualization. in *CHI: Human Factors in Computing Systems*, (1994), 452-458.

Schell, J. and Shochet, J. Designing Interactive Theme Park Rides. *IEEE Computer Graphics & Applications*, *21* (4). 11-13.


3D User Interfaces: Design, Implementation, Usability
CHI 2009
Input devices for 3D UIs
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
30



There are *many* different ways to characterize input devices to be used in 3D UIs, some of which are shown here. In the 3D UI community, researchers often focus on degrees of freedom. But other characteristics can also be important. For example, a typical position tracker provides absolute position information. Some inertial input devices, like the Gyration GyroMouse, which some have seen as a replacement for position trackers, provide relative position information. This difference completely changes the way these devices are used in 3D interaction techniques.



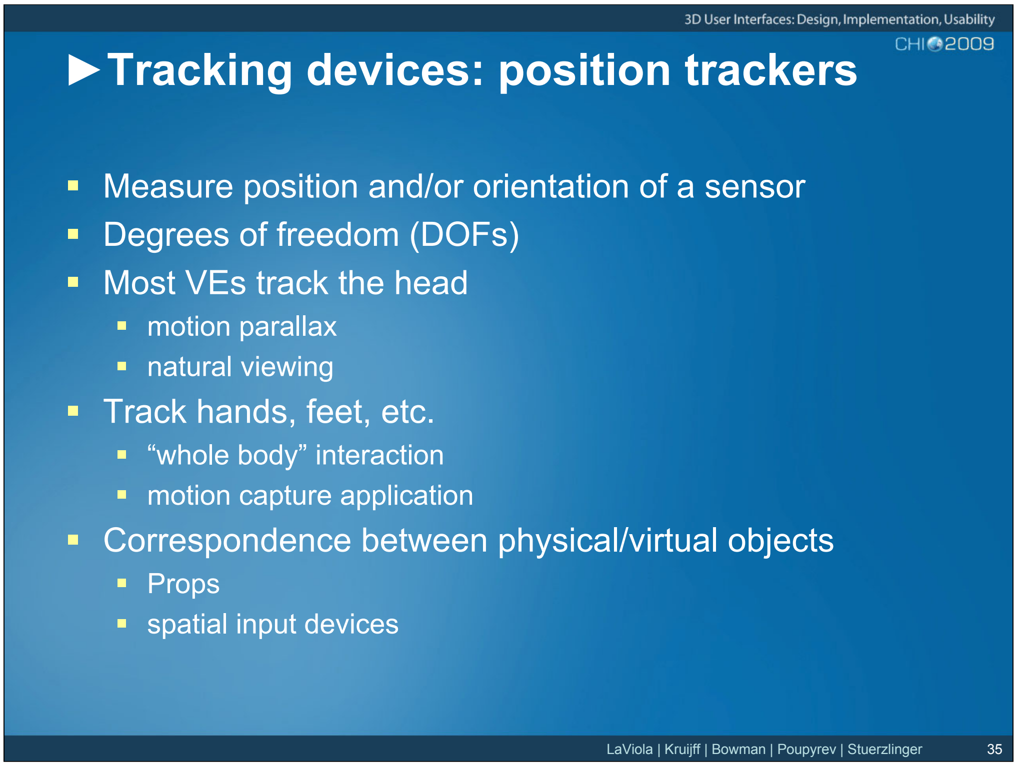
For simplicity, in this lecture, we use a more practical classification system for 3D input devices.

►Desktop devices: 6-DOF devices

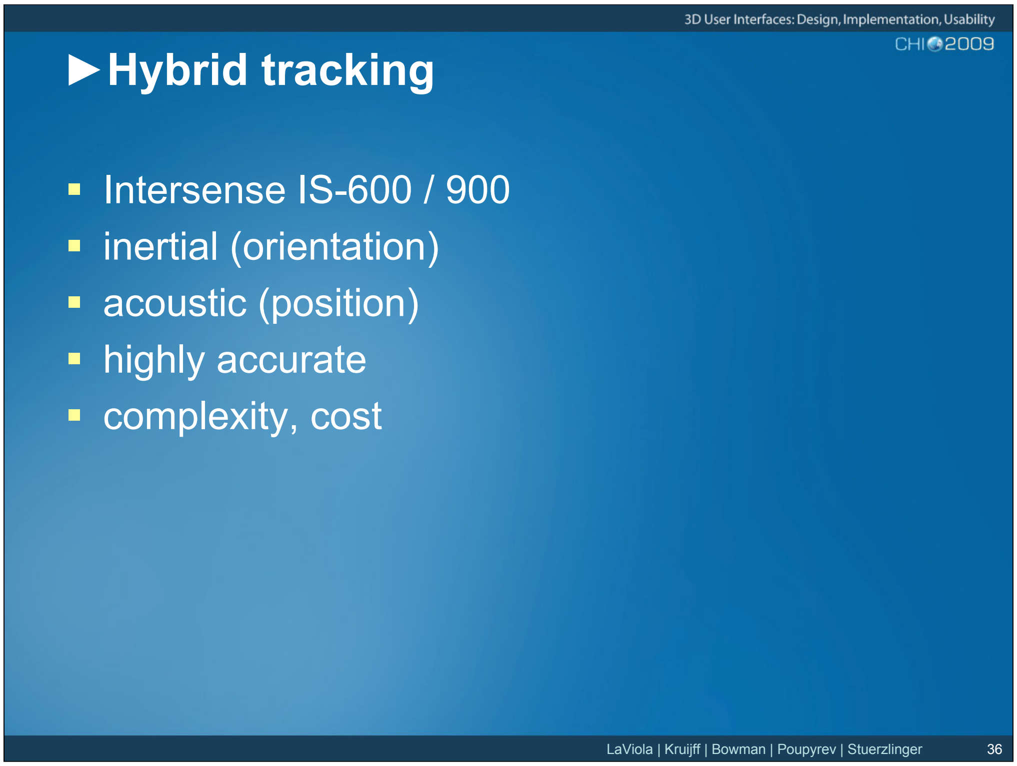
- 6 DOFs without tracking
- Often isometric
- Examples:
  - SpaceBall
  - SpaceMouse
  - SpaceOrb

In the category of desktop devices, the most popular 3D input devices are those that provide six degrees of freedom, such as the SpaceMouse shown here. It allows the user to push/pull/twist the device to specify 3D translation and rotation directly.

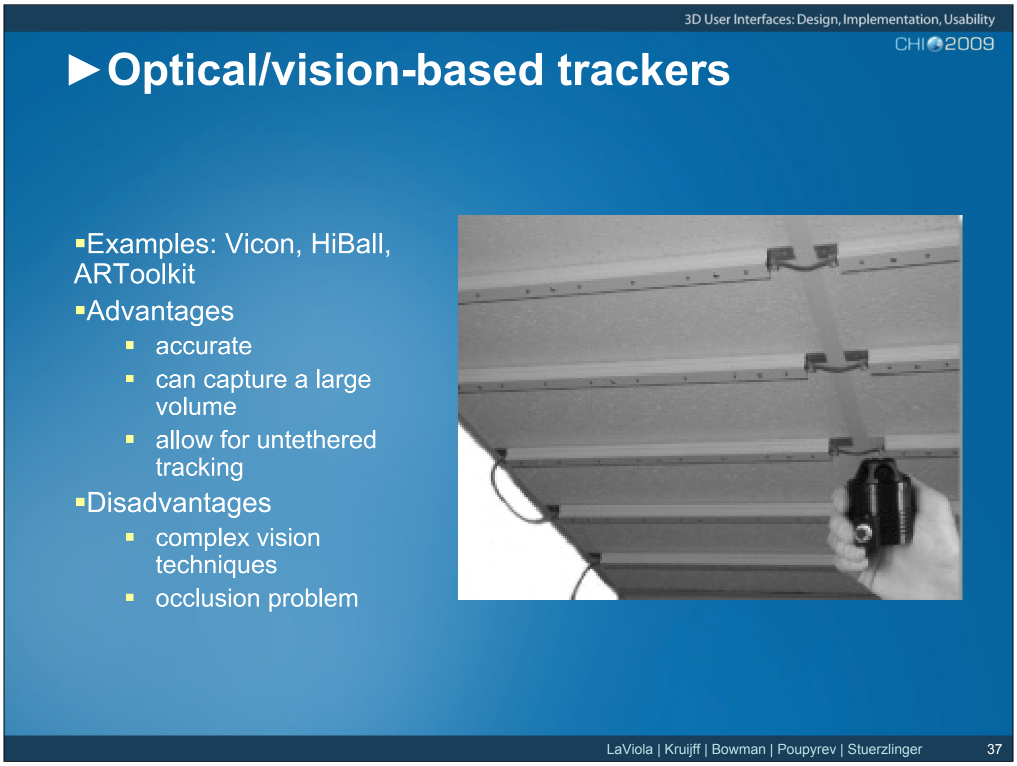
Keyboard input (for text or numeric entry) is often not needed in 3D UIs, but when it is, traditional keyboards are often not practical to use. Thus, 3D UIs often make use of handheld or wearable keyboards, that may use chords instead of individual button presses since they have fewer physical buttons. Soft keyboards, such as those on a TabletPC, may also be used.



Position trackers are on of the most fundamental input devices for 3D UIs. In VEs, they are most often used to track the head and hand(s). But they can also be used to track physical objects that are used as props or spatial input devices (e.g., a physical paintbrush used to paint virtual objects).



One popular type of position tracking today uses a hybrid of inertial tracking for orientation and acoustic (ultrasonic) tracking for position. Such trackers have good accuracy and low latency, and can be wireless. The Intersense IS-900 is a common tracking system of this type.

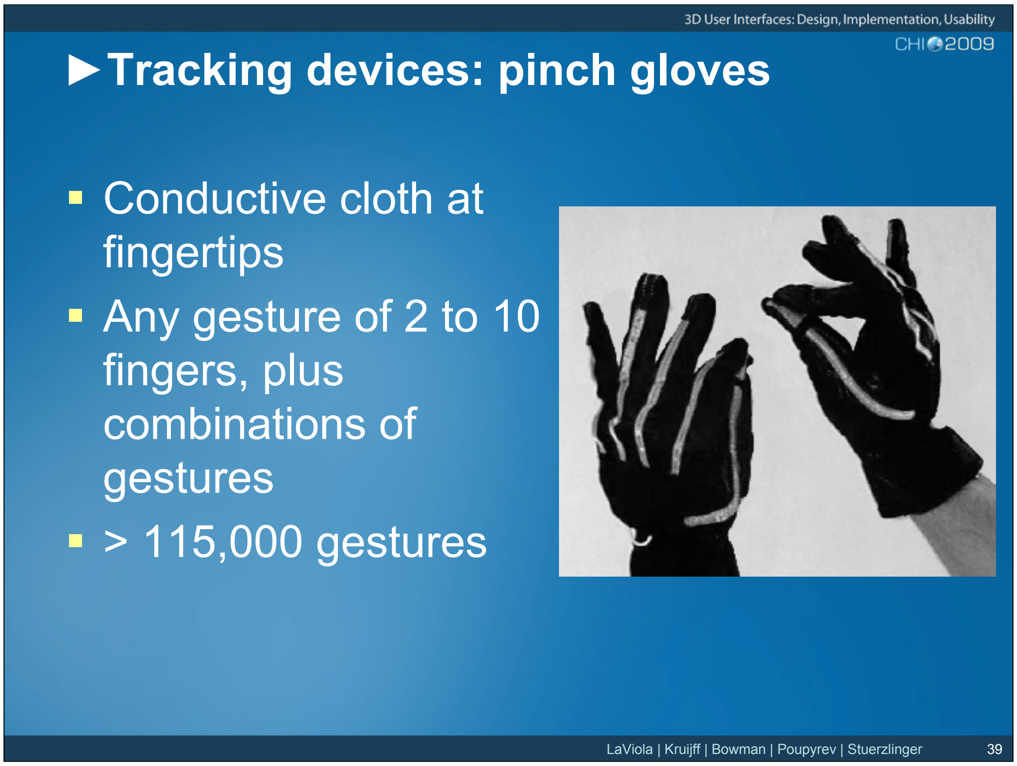


Another popular tracking type for 3D UIs is vision-based tracking. Vicon trackers, which are often used for offline motion capture, can also be used for real-time position tracking. A much lower-cost option is the ARToolkit, which does 6-DOF vision-based tracking using standard webcams and printed tracking markers. The picture shows the HiBall tracking system.

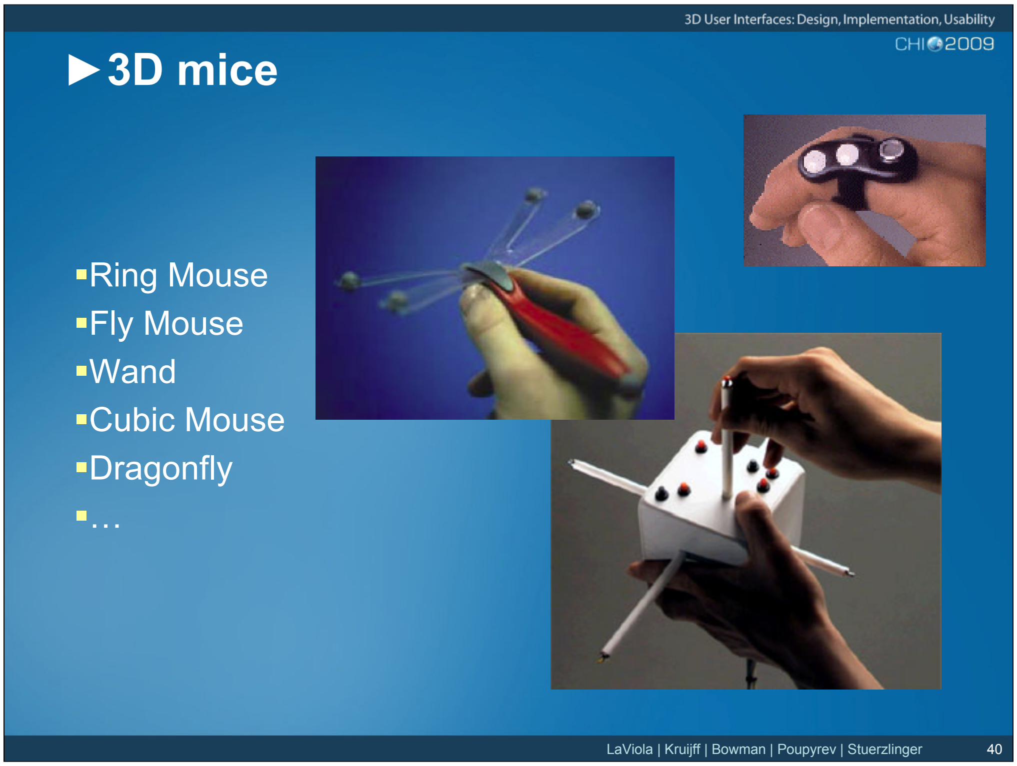
►Tracking devices: bend-sensing gloves

- CyberGlove, 5DT glove
- Reports hand posture
- Gesture:
  - single posture
  - series of postures
  - posture(s) + location or motion
- Calibration issues
- Lack of knowledge on gestural interfaces

Data gloves measure finger movement of the hand by using various kinds of sensor technology. These sensors are embedded in the glove or placed on top of the glove, usually on the back of the hand. The number of sensors in the glove depends on the manufacturer. Virtual Technologies' CyberGlove has either 18 or 22 sensors which can measure at least 2 joints in each finger, wrist roll and yaw, and others. These types of gloves are commonly used for hand gesture and posture recognition which can be applied to a variety of different interface techniques in virtual environments. Fifth Dimension Technologies (5DT) offers gloves that have either 5 sensors, one for each fingertip or 16 sensors, 2 for each finger and abduction between fingers. 5DT also has wireless versions of each glove.



Pinch gloves are a much simpler and more robust glove-based input device for 3D UIs. They do not sense finger movements or postures; rather, they sense when two or more fingers are touching ("pinch gestures"). A large number of gestures are possible, and the gloves can also be tracked to allow spatial input. Pinch gloves are often a good replacement for tracked button devices (flying mice), since the gloves allow many more discrete inputs and don't require the user to hold a device - the hand becomes the device.



The Ring Mouse (top right picture) is a small device worn on the user’s finger which uses ultrasonic tracking. It also has two buttons for generating discrete events. The main advantages of this device is that it is wireless and inexpensive. The Fly Mouse is a 3D mouse that also uses ultrasonic tracking. This device has five buttons instead of two and also can be used as a microphone.

The Cubic Mouse (shown in the figure on the bottom right) is an input device developed at GMD that allows users to intuitively specify three-dimensional coordinates in graphics applications. The device consists of a box with three perpendicular rods passing through the center and buttons for additional input.

The Painting Table is an example of a special-purpose input device that is used in the CavePainting application, a system for painting 3D scenes in a virtual environment. The device uses a set of conductive cloth contacts as well as traditional buttons and digital sliders. Users can dip the paint brush prop into the colored cups to change brush strokes. The bucket is used to throw paint around the virtual canvas.

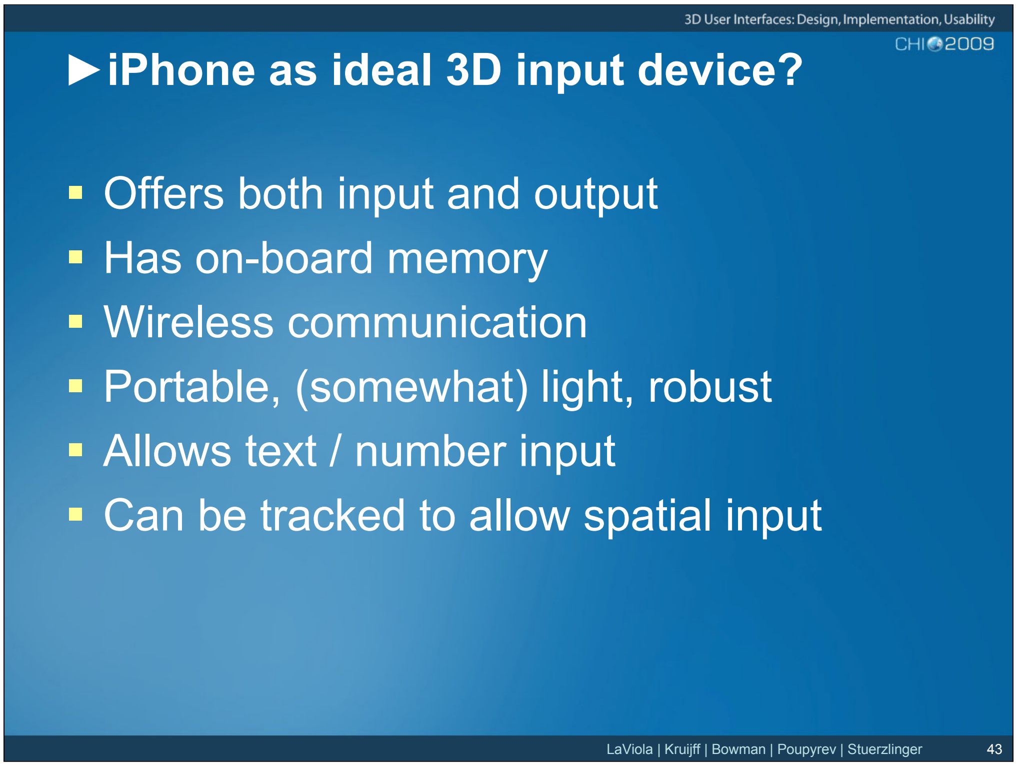
References:

Keefe, D., Acevedo, D., Moscovich, T., Laidlaw, D., and LaViola, J. "CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience", Proceedings of the 2001 Symposium on Interactive 3D Graphics, 85-93, 2001.

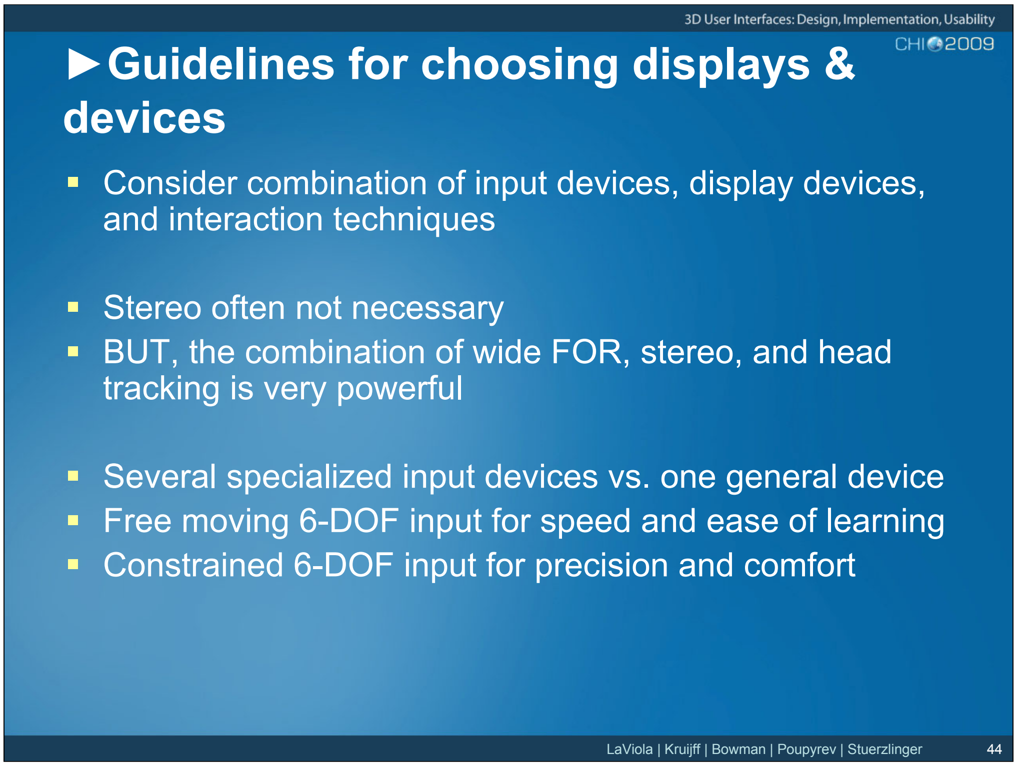
▶Direct human input

- Physiological signals
- Eye tracking
- Brain-computer interfaces

The human body and brain are also sources of input for 3D UIs. In particular, brain-computer interfaces (BCIs) have great potential for 3D UI input.

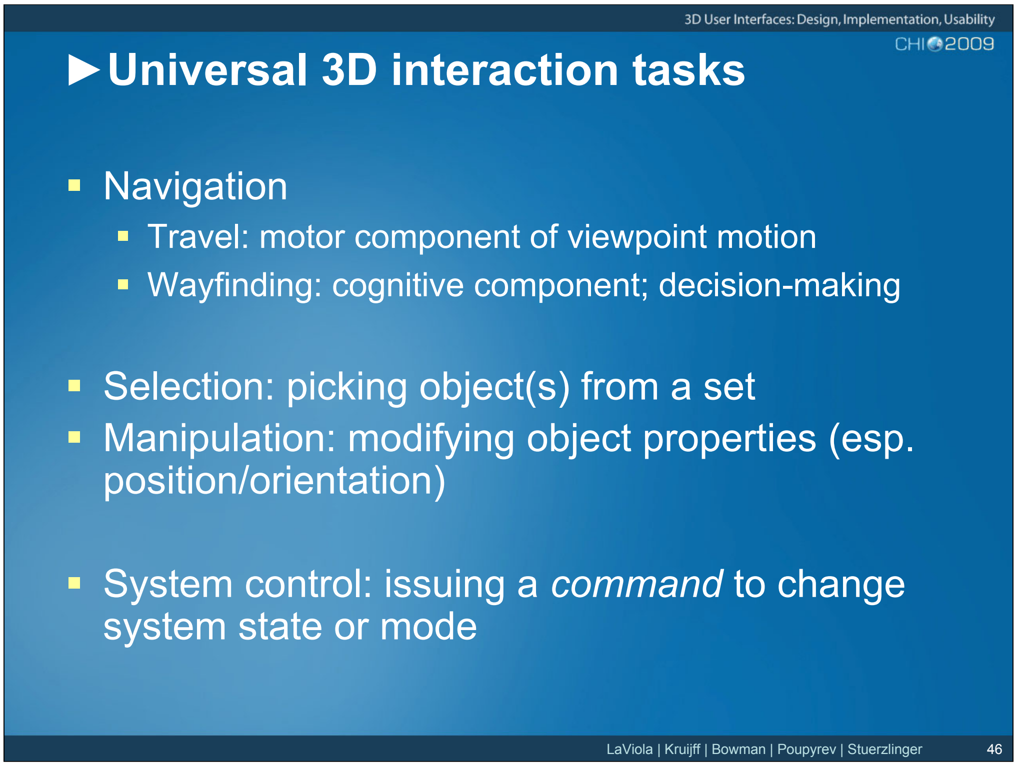


Many researchers have used PDAs or tabletPCs for input in 3D UIs, for the reasons shown. They provide several advantages, and overcome some of the common usability problems in 3D UIs (e.g., it's difficult to provide menus or readable text on 3D displays).



Choosing displays and input devices for 3D UIs is difficult because of the wide range of technologies available, and the lack of standards. In addition, since input devices don't determine interaction techniques, the techniques must also be considered when choosing devices.

Basic 3D interaction techniques

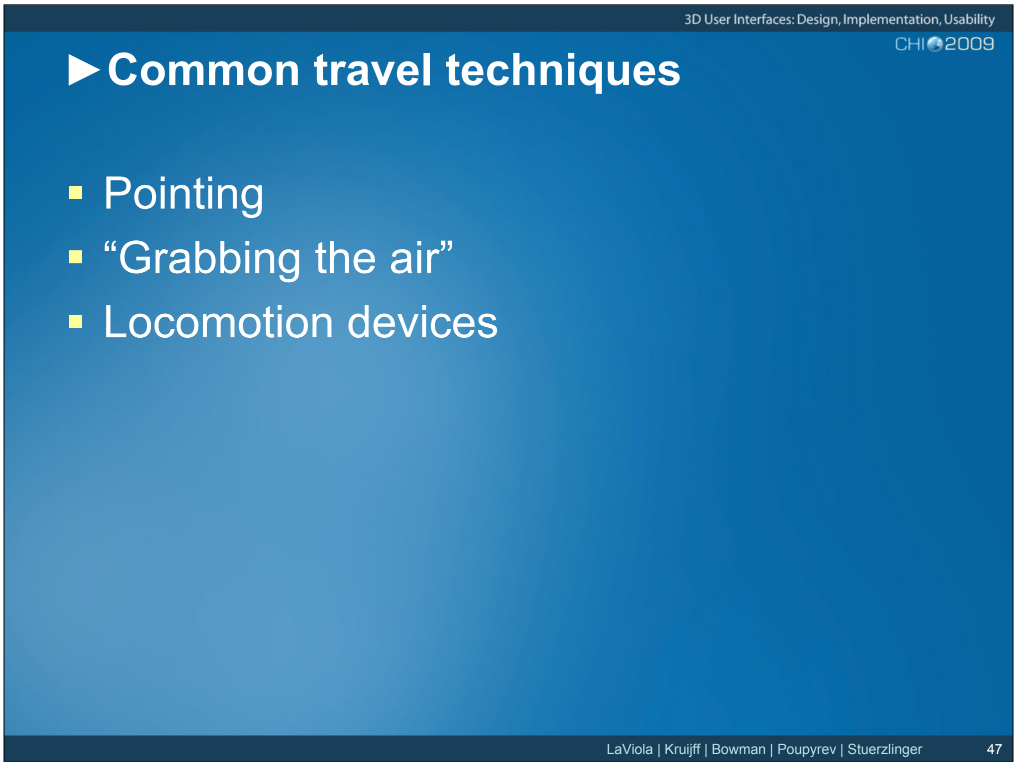


We'll be discussing techniques for four basic 3D interaction tasks that are found in most complex 3D applications Obviously, there are other tasks which are specific to an application domain, but these are some basic building blocks that can often be combined to create a more complex task.

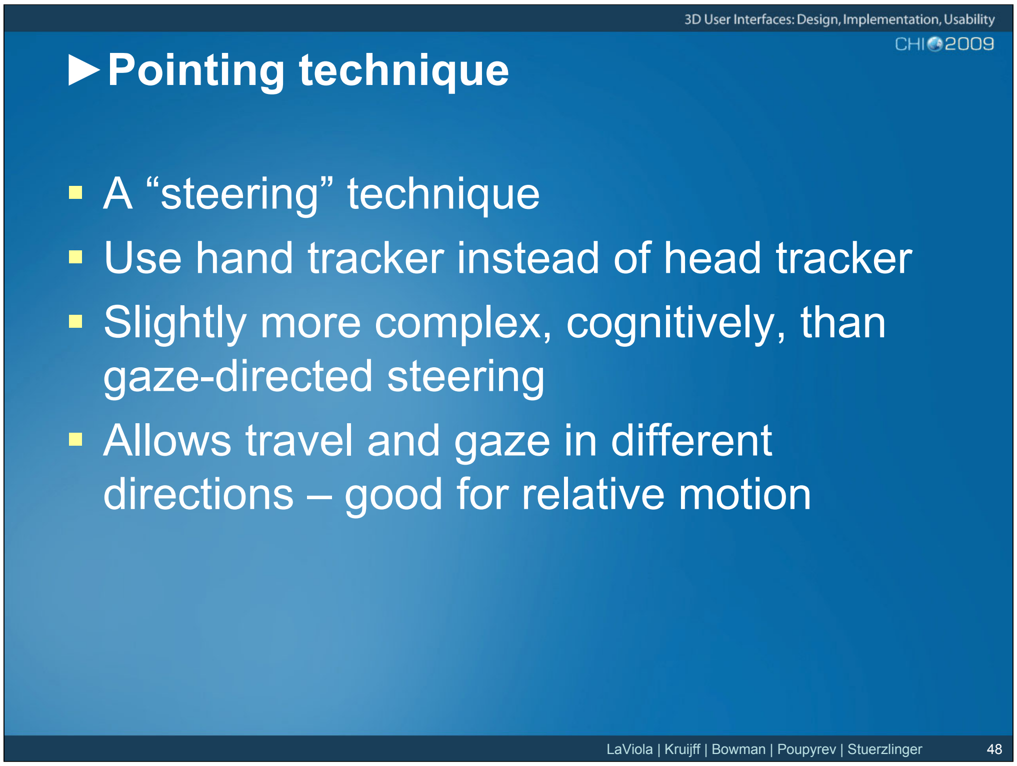
Navigation is the most common VE task, and is actually composed of two tasks. Travel is the motor component of navigation, and just refers to the physical movement from place to place. Wayfinding is the cognitive or decision-making component of navigation, and it asks the questions, "where am I?", "where do I want to go?", "how do I get there?", and so on.

Selection is simply the specification of an object or a set of objects for some purpose. Manipulation refers to the specification of object properties (most often position and orientation, but also other attributes). Selection and manipulation are often used together, but selection may be a stand-alone task. For example, the user may select an object in order to apply a command such as "delete" to that object.

System control is the task of changing the system state or the mode of interaction. This is usually done with some type of command to the system (either explicit or implicit). Examples in 2D systems include menus and command-line interfaces. It is often the case that a system control technique is composed of the other three tasks (e.g. a menu command involves selection), but it's also useful to consider it separately since special techniques have been developed for it and it is quite common.



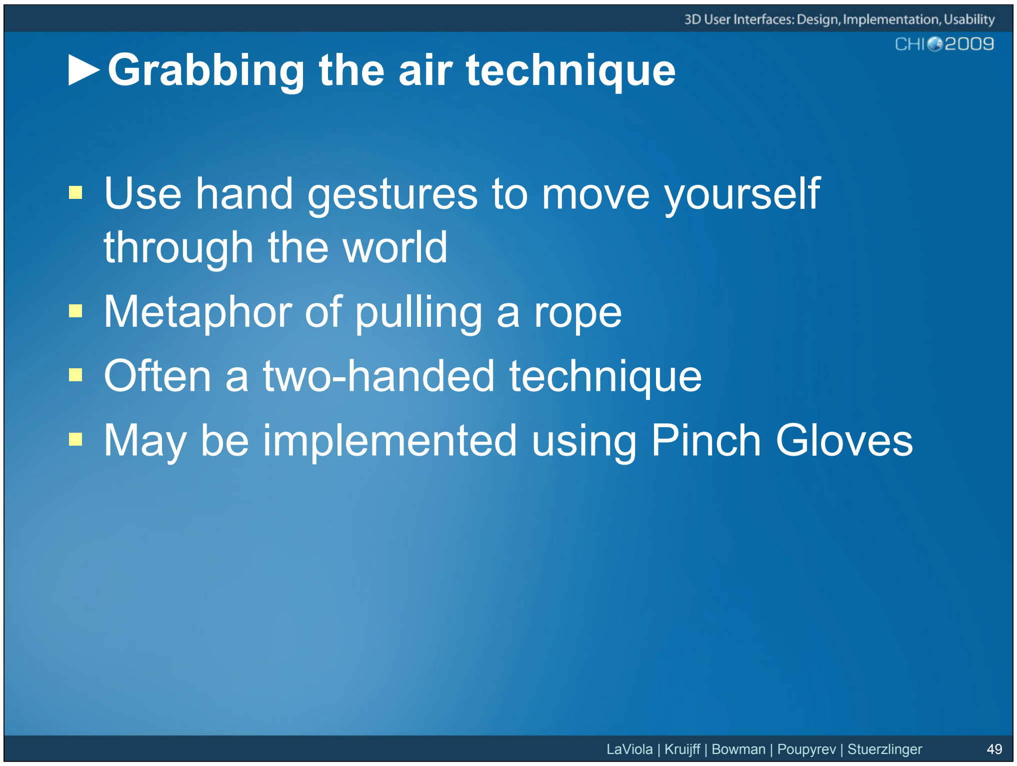
We’ll discuss three common techniques, focusing on innovative techniques beyond what is normally seen in desktop 3D UIs.



Pointing is a steering technique (where the user continuously specifies the direction of motion). In this case, the hand’s orientation is used to determine direction. This technique is somewhat harder to learn for some users, but is more flexible than gaze-directed steering.

See: Mine, M. (1995). *Virtual Environment Interaction Techniques* (Technical Report TR95-018): UNC Chapel Hill CS Dept., and

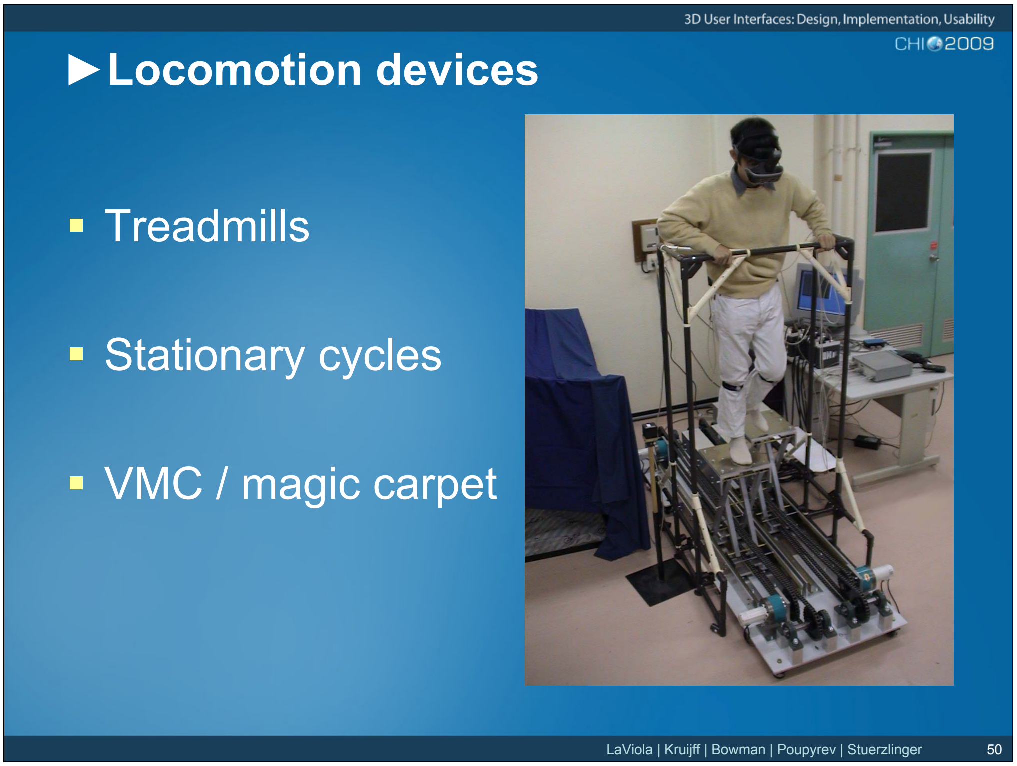
Bowman, D. A., Koller, D., & Hodges, L. F. (1997). *Travel in Immersive Virtual Environments: an Evaluation of Viewpoint Motion Control Techniques.* Proceedings of the Virtual Reality Annual International Symposium, 45-52.



The “grabbing the air” technique uses the metaphor of literally grabbing the world around you (usually empty space), and pulling yourself through it using hand gestures. This is similar to pulling yourself along a rope, except that the “rope” exists everywhere, and can take you in any direction.

This technique may be done with one or two hands, and is often implemented using Pinch Gloves™.

See: Mapes, D., & Moshell, J. (1995). A Two-Handed Interface for Object Manipulation in Virtual Environments. *Presence: Teleoperators and Virtual Environments, 4*(4), 403-416.



Instead of relying solely on common input devices and software-based interaction techniques, locomotion devices are special-purpose devices specifically designed for the task of travel. These can range from simple exercise bikes, to omni-directional treadmills.

►Classification of travel and locomotion

	Virtual turning	Real turning
Virtual translation	Desktop VEs Vehicle simulators CAVE wand	Most HMD systems Walking in place Magic Carpet
Real translation	Stationary cycles Treadport Biport	Wide-area tracking UNIPORT ODT

A useful way to classify locomotion devices and other travel techniques is their use of virtual and physical movements - both translation and rotation. We know that physical movements can be helpful in helping users maintain spatial orientation, although providing both real translation and real turning can be costly and difficult.

▶Travel design guidelines

- Make simple travel tasks simple (target-based techniques for motion to an object, steering techniques for search).
- Provide multiple travel techniques to support different travel tasks in the same application.
- Use graceful transitional motions if overall environment context is important.
- Train users in sophisticated strategies to help them acquire survey knowledge.
- Consider integrated (“cross-task”) ITs if travel is used in the context of another task (e.g. manipulation).

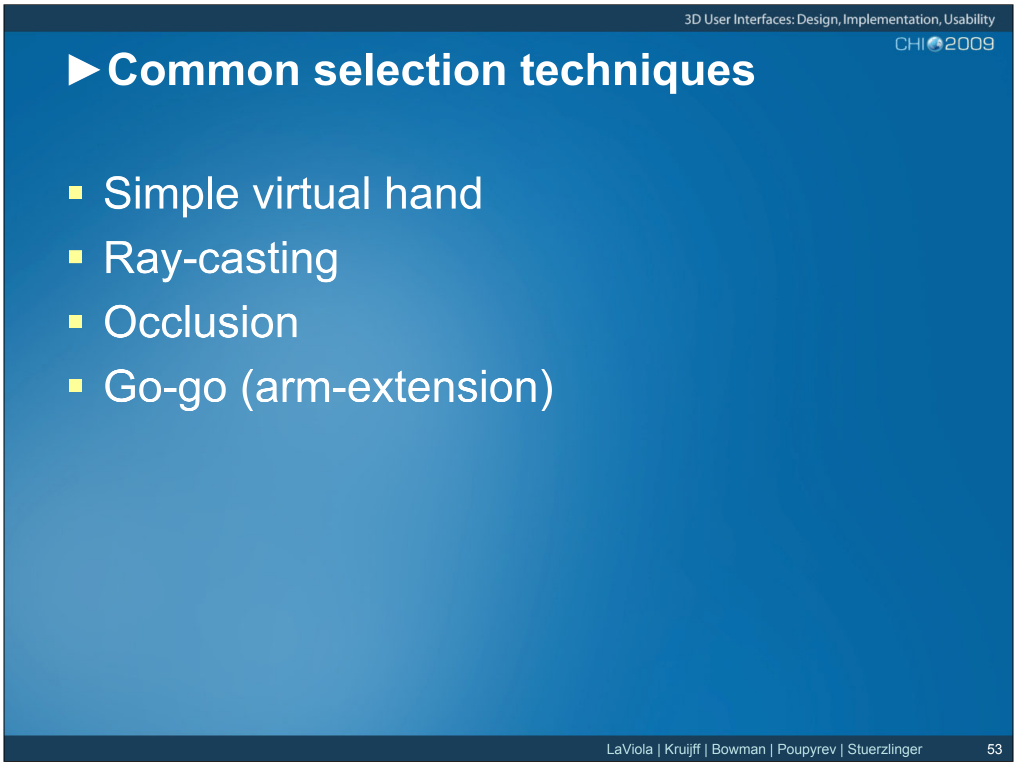
Most travel tasks are simple in the mind of the user – they just want to change their location while focusing on something else. Thus, you should use a technique that meets the requirements of the task: e.g. use a target-based technique if the only goal is to move between known objects - don’t put unnecessary cognitive load on the user.

Remember the differences between tasks such as exploration and primed search – you may need more than one technique. There is a tradeoff between the specificity of the technique and the amount of learning load you want to put on the user. In many cases, multiple techniques requiring a bit more learning time may be much more efficient in the long run.

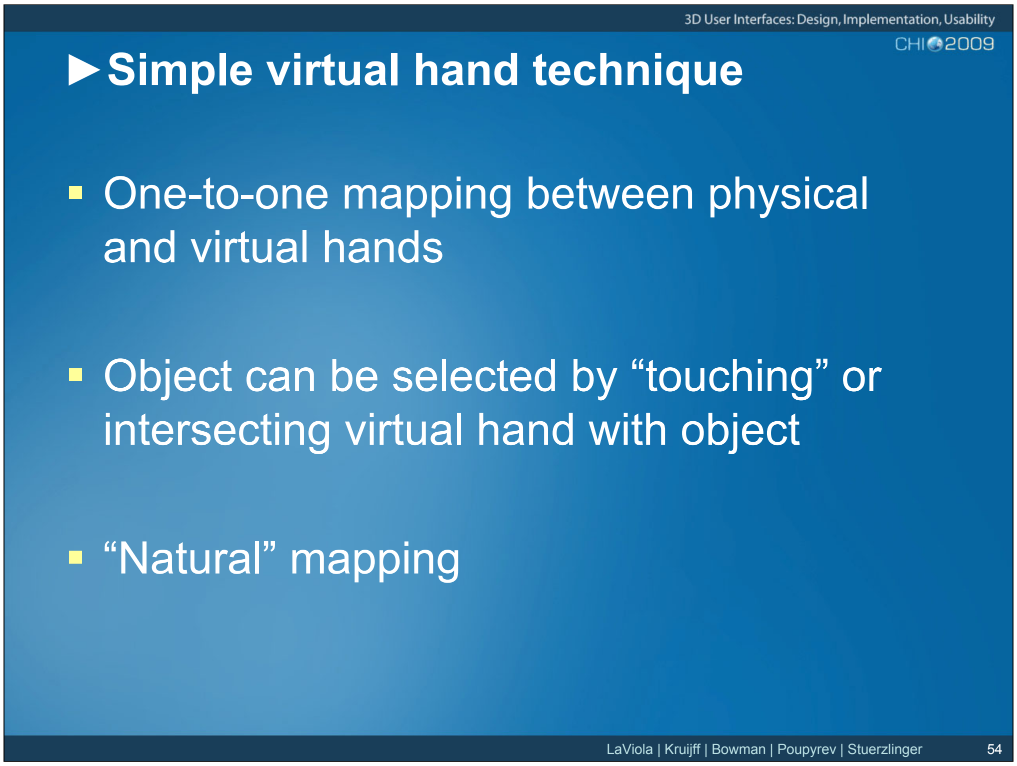
Many applications require the user to be aware of their location within the space, have an overall survey knowledge of the space, etc. (see the lecture on wayfinding). In these cases it is important to use transitional motion between locations, even if it is fast, in order to maintain awareness of the space. (A good use of this concept in a desktop system is Mackinlay, Card, and Robertson, Rapid controlled movement through a virtual 3D workspace, SIGGRAPH ’90, 171-176.)

Strategies (how the user uses the technique) are as important as the technique itself, especially in tasks requiring spatial knowledge. Therefore, you should provide training, instructions, and guidance to help the user take advantage of the technique.

Cross-task ITs can be useful if travel is not the main interaction, but is only used, for example, to gain a better viewpoint on a manipulation task. Remember that such motion can be tiring, however, and should not be used for very long exposure period applications.

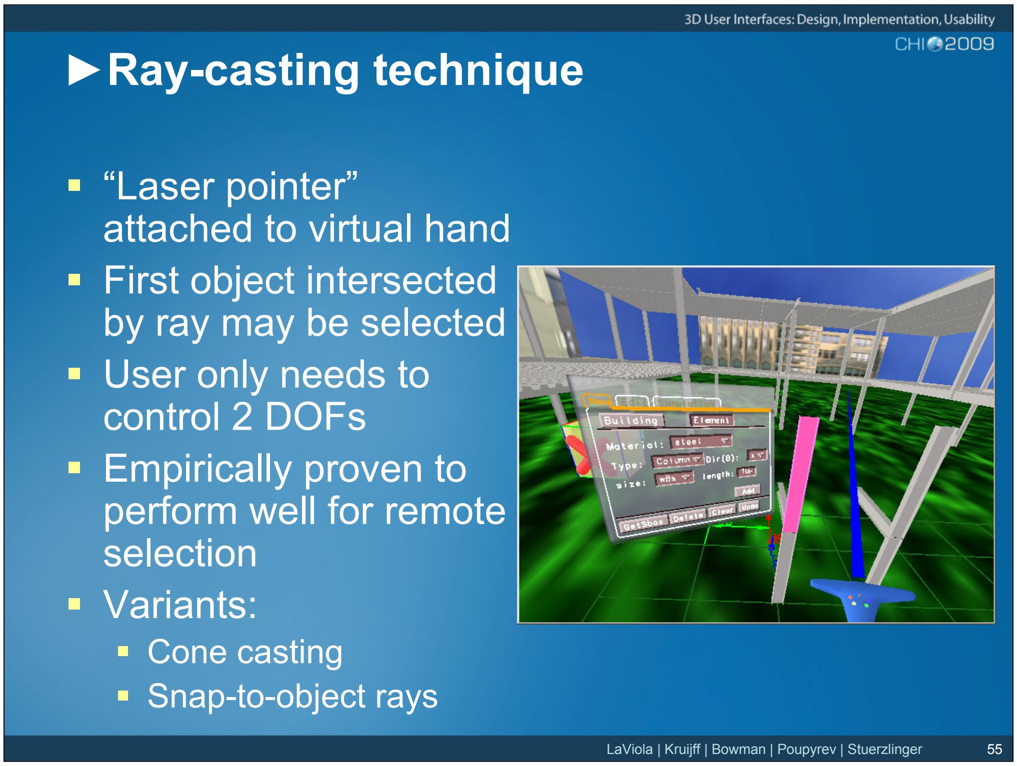


We'll discuss four selection techniques, again focusing on techniques that use 3D input devices.



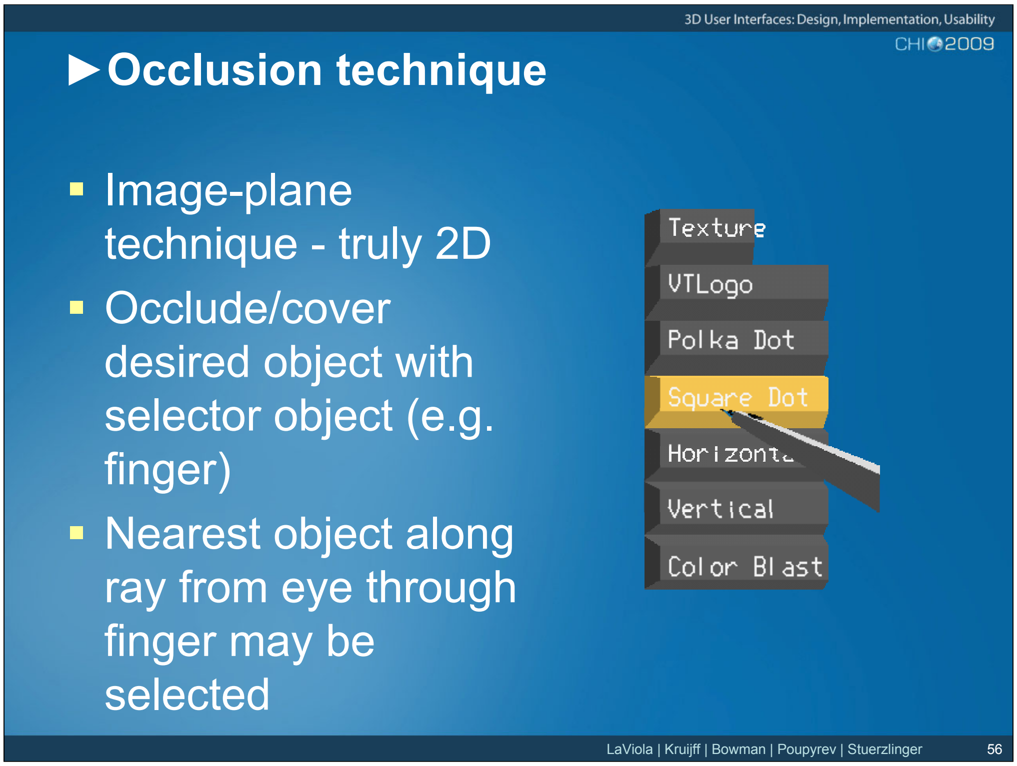
The most common technique is the simple virtual hand, which does “real-world” selection via direct “touching” of virtual objects. In the absence of haptic feedback, this is done by intersecting the virtual hand (which is at the same location as the physical hand) with a virtual object.

Implementing this technique is simple, provided you have a good intersection/collision algorithm. Often, intersections are only performed with axis-aligned bounding boxes or bounding spheres rather than with the actual geometry of the objects.



Another common technique is ray-casting. This technique uses the metaphor of a laser pointer – an infinite ray extending from the virtual hand. The first object intersected along the ray is eligible for selection. This technique is efficient, based on experimental results, and only requires the user to vary 2 degrees of freedom (pitch and yaw of the wrist) rather than the 3 DOFs required by the simple virtual hand and other location-based techniques.

See: Mine, M. (1995). *Virtual Environment Interaction Techniques* (Technical Report TR95-018): UNC Chapel Hill CS Dept.

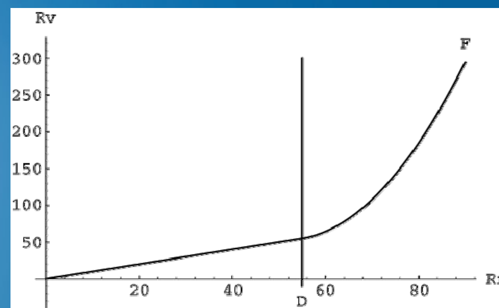


Next, we'll cover the occlusion technique (also called the "sticky finger" technique). This technique works in the plane of the image – that is, you select an object by "covering" it with the virtual hand so that it is occluded from your point of view. Geometrically, this means that a ray is emanating from your eye, going through your finger, and then intersecting an object.

See: Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., & Mine, M. (1997). *Image Plane Interaction Techniques in 3D Immersive Environments.* Proceedings of the ACM Symposium on Interactive 3D Graphics, 39-44.

►Go-Go technique

- Arm-extension technique
- Like simple v. hand, touch objects to select them
- Non-linear mapping between physical and virtual hand position
- Local and distant regions



The Go-Go technique is based on the simple virtual hand, but it introduces a non-one-to-one mapping between the physical hand and the virtual hand, so that the user’s reach is greatly extended. This is called an arm-extension technique.

The graph shows the mapping between the physical hand distance from the body on the x-axis and the virtual hand distance from the body on the y-axis. There are two regions. When the physical hand is at a depth less than a threshold ‘D’, the one-to-one mapping applies. Outside D, a non-linear mapping is applied, so that the farther the user stretches, the faster the virtual hand moves away.

See: Poupyrev, I., Billinghurst, M., Weghorst, S., & Ichikawa, T. (1996). *The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR.* Proceedings of the ACM Symposium on User Interface Software and Technology, 79-80.

3D User Interfaces: Design, Implementation, Usability

CHI 2009

▶ Common manipulation techniques

- Simple virtual hand
- HOMER
- Scaled-world grab
- World-in-miniature

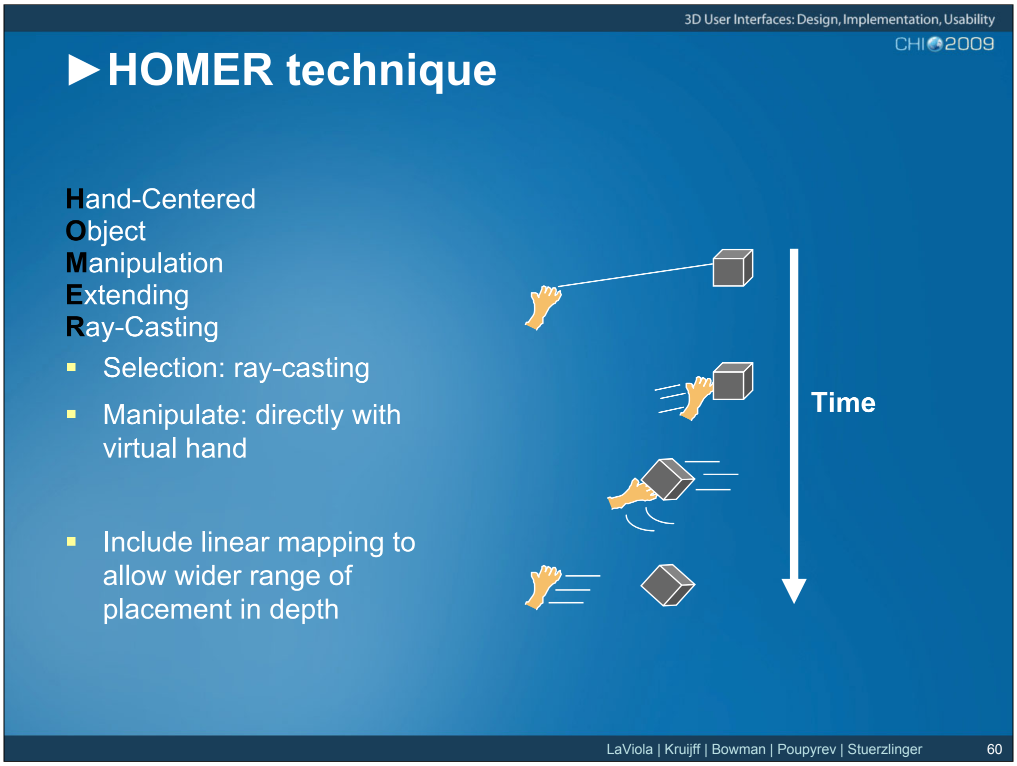
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger 58

We'll discuss four 3D object manipulation techniques.

We already saw the simple virtual hand technique for selection. When this technique is used for object manipulation, the implementation is quite easy. It simply involves making a change to the scene graph by attaching the selected object to the virtual hand. Then, as the virtual hand moves and rotates, the selected object will inherit those transformations. When the object is released, it should just be reattached to its earlier location in the tree.

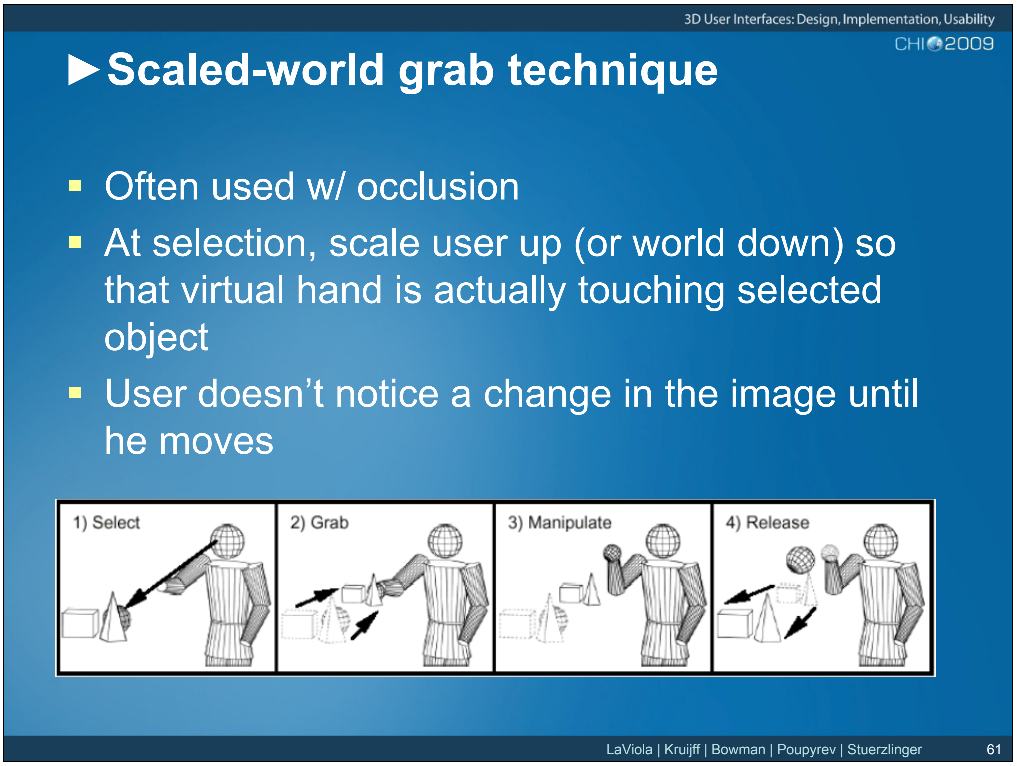
The only tricky issue here is that you must ensure when grabbing or releasing the object that it does not move (in the world CS). If you simply make the object a child of the hand, it may move since its position is now being interpreted relative to a new CS (the hand's). To be completely general, then, you must get the object's position *p* in the world CS first, then do the attachment, then calculate *p*'s location in the hand CS, then move the object to that position (relative to the hand). The opposite transformation is done upon release.

This same basic procedure works for other techniques that simply attach the object to the selector, like Go-Go and ray-casting.



The HOMER technique uses ray-casting for selection and then moves the virtual hand to the object for hand-centered manipulation. The depth of the object is based on a linear mapping. The initial torso-physical hand distance is mapped onto the initial torso-object distance, so that moving the physical hand twice as far away also moves the object twice as far away. Also, moving the physical hand all the way back to the torso moves the object all the way to the user's torso as well.

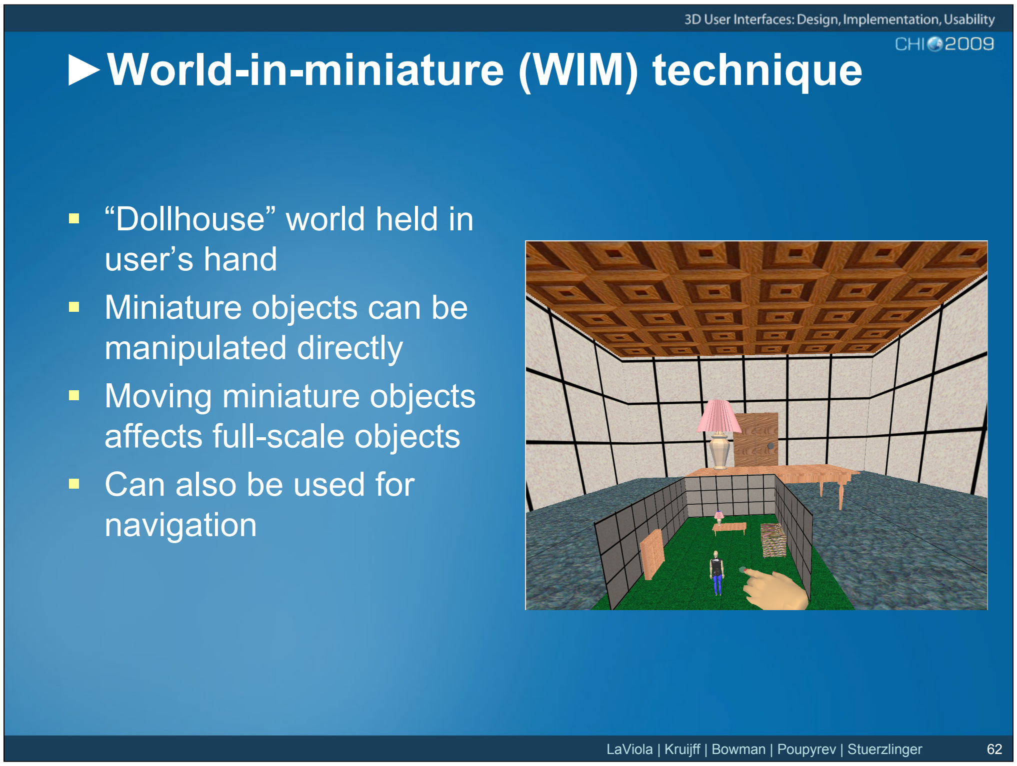
See: Bowman, D., & Hodges, L. (1997). *An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments.* Proceedings of the ACM Symposium on Interactive 3D Graphics, 35-38.



The scaled-world grab technique is often used with occlusion selection. The idea is that since you are selecting the object in the image plane, you can use the ambiguity of that single image to do some magic. When the selection is made, the user is scaled up (or the world is scaled down) so that the virtual hand is actually touching the object that it was occluding. If the user doesn't move (and the graphics are not stereo), there is no perceptual difference between the images before and after the scaling. However, when the user starts to move the object and/or his head, he realizes that he is now a giant (or that the world is tiny) and he can manipulate the object directly, just like the simple virtual hand.

See: Mine, M., Brooks, F., & Sequin, C. (1997). *Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction.* Proceedings of ACM SIGGRAPH, 19-26, and

Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., & Mine, M. (1997). *Image Plane Interaction Techniques in 3D Immersive Environments.* Proceedings of the ACM Symposium on Interactive 3D Graphics, 39-44.



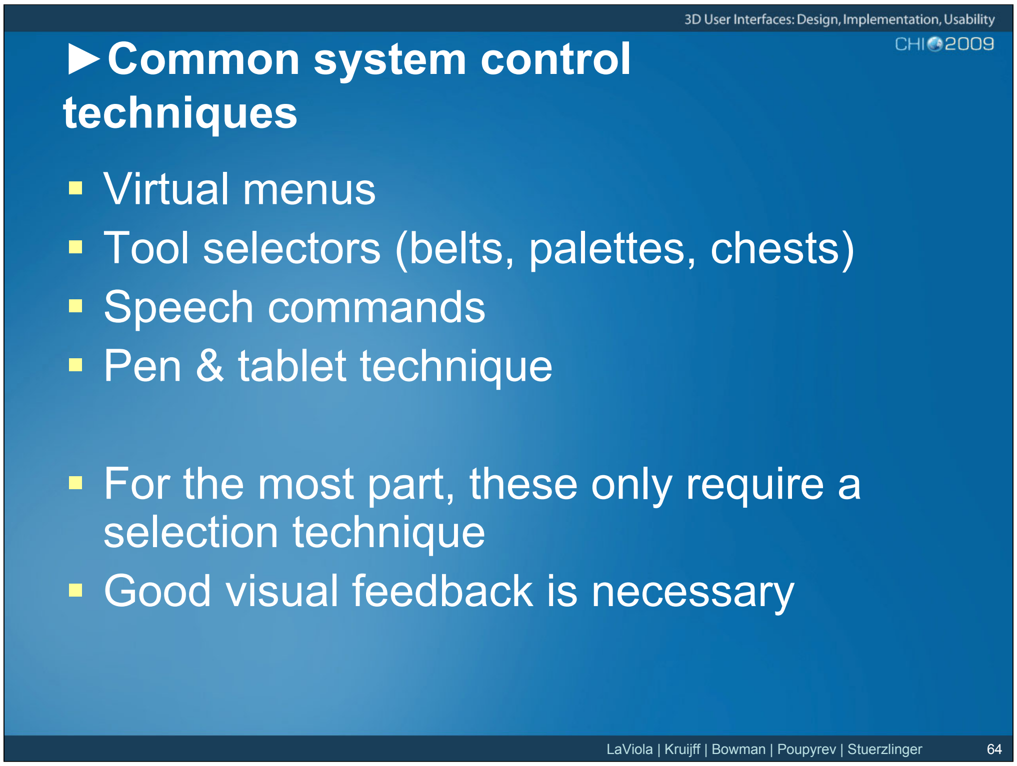
The world-in-miniature (WIM) technique uses a small “dollhouse” version of the world to allow the user to do indirect manipulation of the objects in the environment. Each of the objects in the WIM is selectable using the simple virtual hand technique, and moving these objects causes the full-scale objects in the world to move in a corresponding way. The WIM can also be used for navigation by including a representation of the user, in a way similar to the map-based travel technique, but including the 3rd dimension.

See: Stoakley, R., Conway, M., & Pausch, R. (1995). *Virtual Reality on a WIM: Interactive Worlds in Miniature.* Proceedings of CHI: Human Factors in Computing Systems, 265-272, and

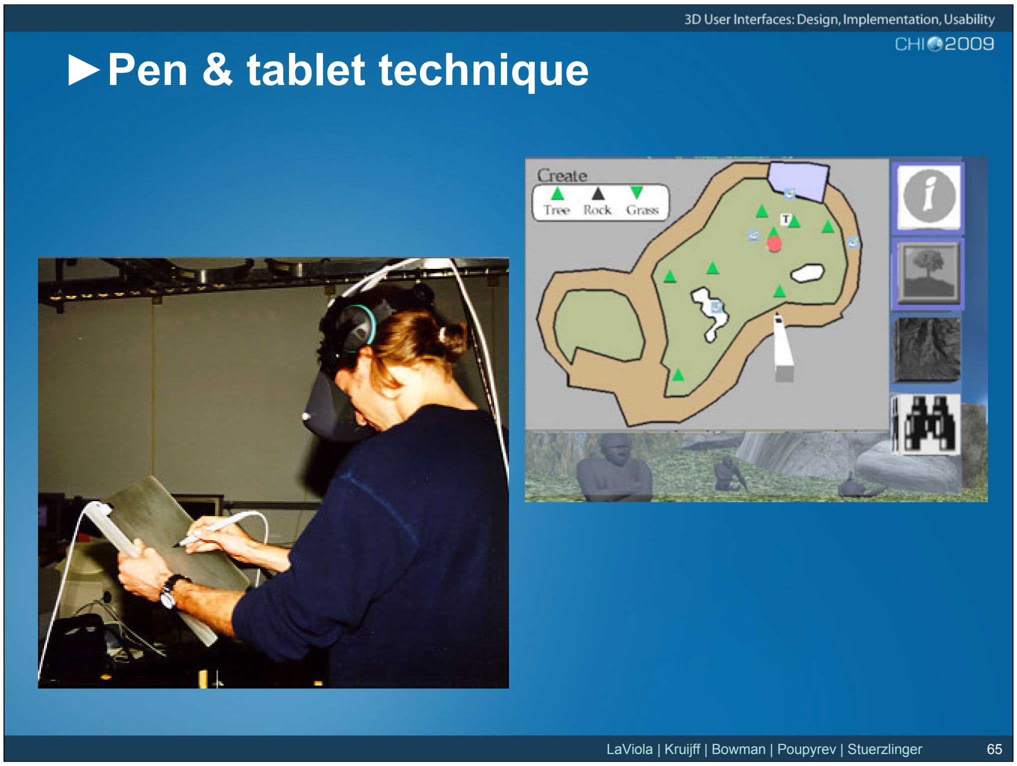
Pausch, R., Burnette, T., Brockway, D., & Weiblen, M. (1995). *Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures.* Proceedings of ACM SIGGRAPH, 399-400.

▶ Manipulation design guidelines

- Match the interaction technique to the device
- Use techniques that can help to reduce clutching
- Use pointing techniques for selection and virtual hand techniques for manipulation
- Reduce degrees of freedom when possible



System control is a wide-ranging topic, and there are many different techniques, some of which are listed here. For the most part, these techniques are not difficult to implement, since they mostly involve selection, which we've already covered. For example, virtual menu items might be selected using ray-casting. For all of the techniques, good visual feedback is required, since the user needs to know not only what he is selecting, but what will happen when he selects it.

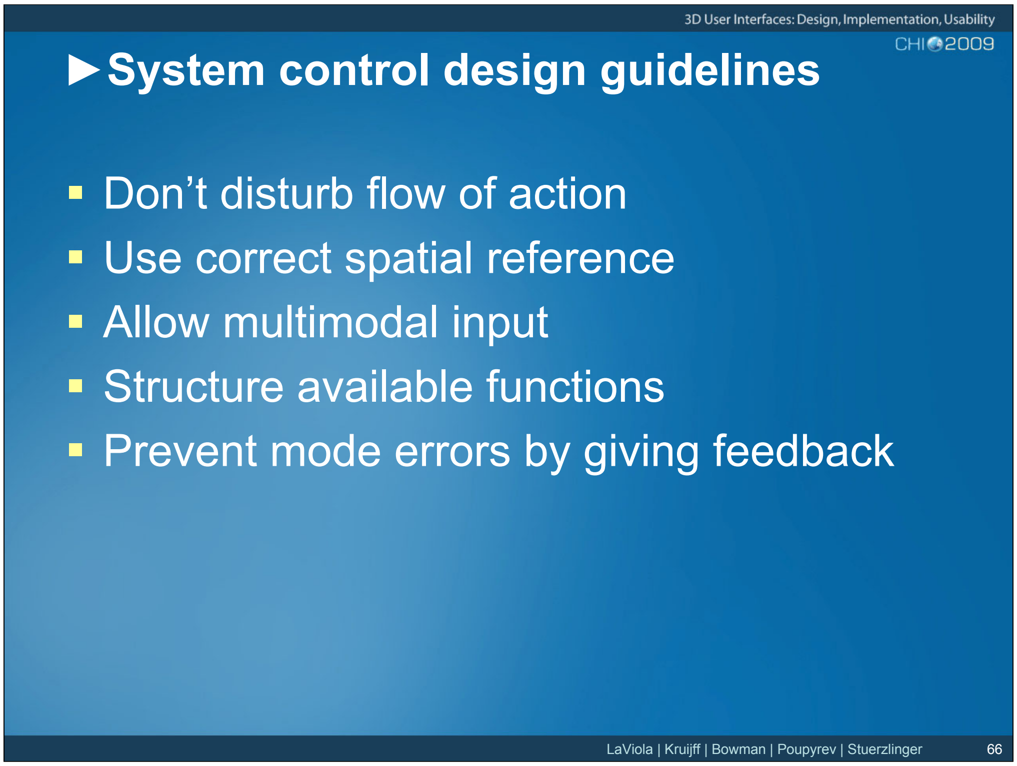


I only want to touch on one system control technique, because of its widespread use. The pen & tablet technique uses a physical pen and tablet (see left image). In the virtual world, the user sees a virtual pen and tablet, and a 2D interface on the surface of the virtual tablet (right image). The physical devices provide near-field haptics and constraints that make such an interface easy to use.

As we mentioned in the section on input devices, the same effect (and more) can be achieved with a tabletPC, but this only works if your display device allows the user to see the physical world (i.e., it wouldn't work with an HMD).

See: Angus, I., & Sowizral, H. (1995). *Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment.* Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems, 282-293, and

Schmalsteig, D., Encarnacao, L., & Szalzvari, Z. (1999). *Using Transparent Props For Interaction with The Virtual Table.* Proceedings of the ACM Symposium on Interactive 3D Graphics, 147-154.

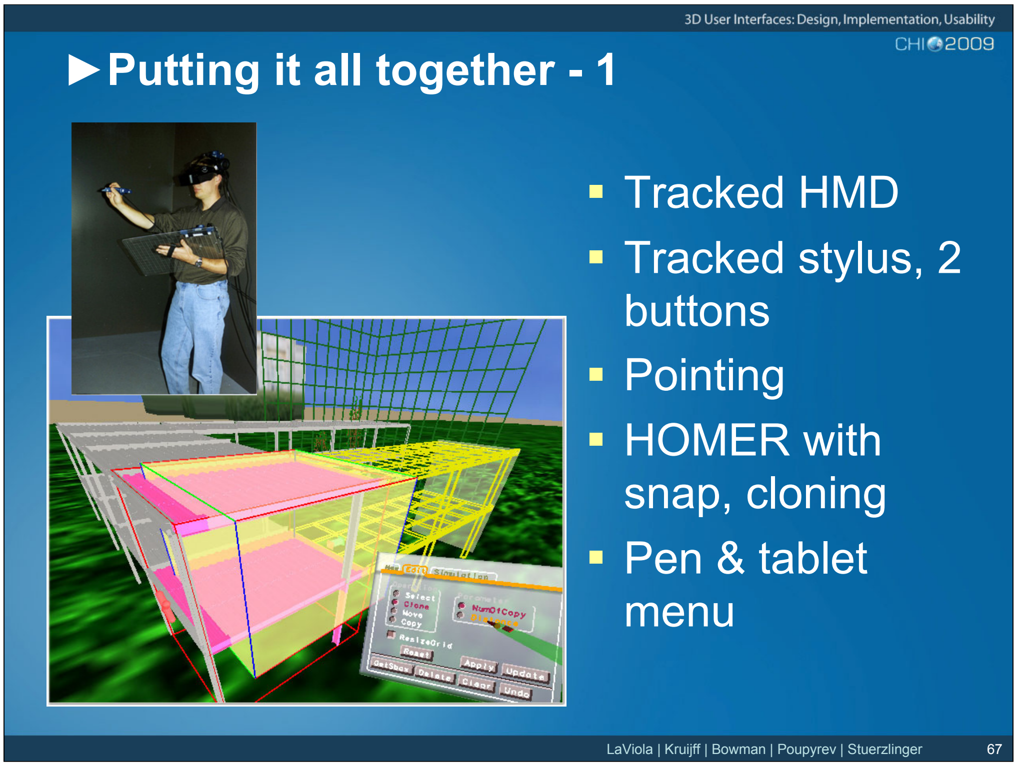


Extracted from the descriptions of system control techniques, several important design guidelines can be stated. Due to the relative lack of formal evaluations, these guidelines are primarily based on tendencies described by researchers and personal experience.

System control is often integrated within another universal interaction task. Due to this integration, we should avoid disturbing the flow of action of an interaction task. The user should stay focused on the task. "Modeless" interaction (where the mode changes are very natural) is ideal. One way of supporting the user to easily access a system control interface is by using a correct spatial reference. This guideline is of course mostly applicable to graphical menus, but tools also benefit from a strong spatial reference. Another method to allow a more seamless integration of system control into a flow of action is to use a multimodal, or hybrid, system control interface. Multimodal interfaces can increase the performance of issuing a command, and may allow multiple channels to access the system control interface. However, keep in mind that multimodal system control is not always suitable or applicable.

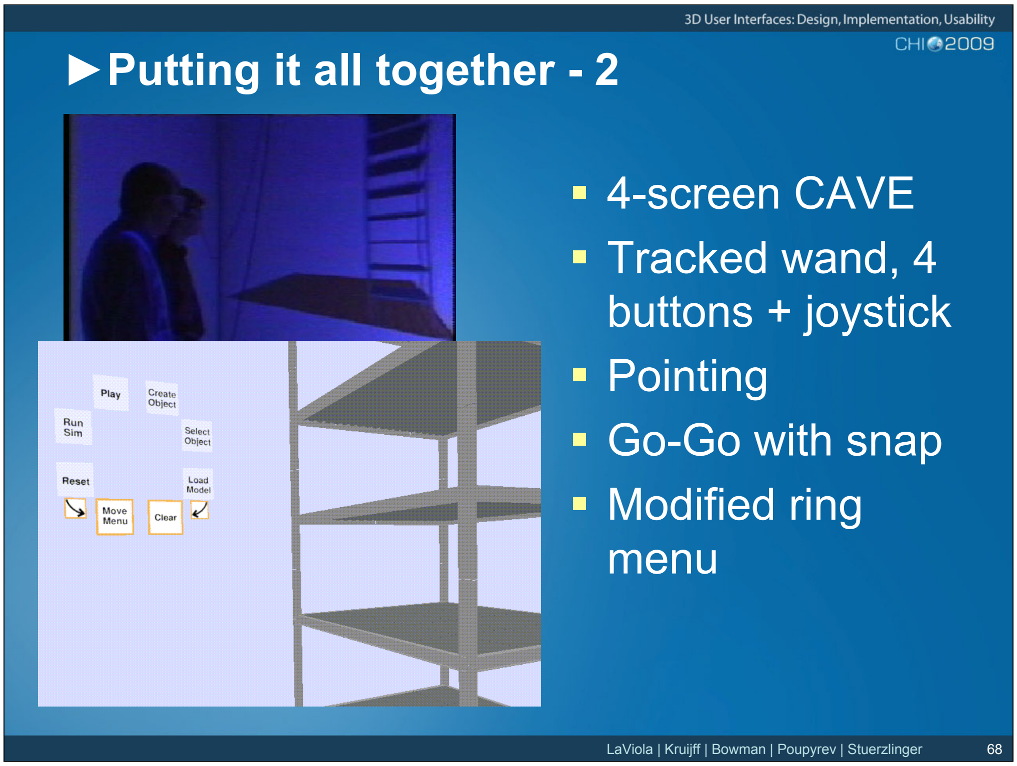
After the user has accessed a system control interface, he/she has to select an item from a set: when this set is large, i.e. when a large number of functions are available, one needs to structure the items. As stated in the guidelines on graphical menus, this might be achieved by methods like using context-sensitivity, or by clearly communicating the hierarchy of items and (sub)menus.

Finally, always try to prevent mode errors by providing the user with appropriate feedback during and after selection of a command. Mode errors can be highly disturbing and they interrupt the flow of action in an application.



I want to conclude with three examples showing complete 3D UIs. All of the 3D UIs are for the same application, called Virtual-SAP. The application allows structural engineers (and engineering students) to construct 3D building structures in a virtual environment.

The first 3D UI uses a fairly standard HMD setup. Because HMD users can't see other devices, we used the virtual pen & tablet approach for system control, with corresponding physical props. The pen can also be used to fly through the world (with the pointing technique), and to select and manipulate objects (with the HOMER technique).



The second 3D UI for Virtual-SAP used a CAVE as the display device. The pen & tablet technique is more difficult to do in the CAVE, so we created a new system control technique with a circular menu. Instead of making users point to the menu items, we use two buttons on the input device to rotate the menu in either direction, and two other buttons to select items that are in the bottom two “bins” of the menu. This is fast and accurate. Ray-casting (for the HOMER technique) was also less usable in the CAVE because of difficulty seeing the ray in stereo, so we used a modified version of the Go-Go technique with a snapping feature for precision.

Finally, we wanted to use Virtual-SAP on a portable VR system in classrooms. So we chose an inexpensive consumer HMD and a simple 3DOF orientation tracker that could be used anywhere. This meant we couldn't track the hand, so we used a chord keyboard device with 12 buttons. This led to a "remote control" metaphor for the menu, and travel, selection, and manipulation techniques based on head orientation rather than hand movements.