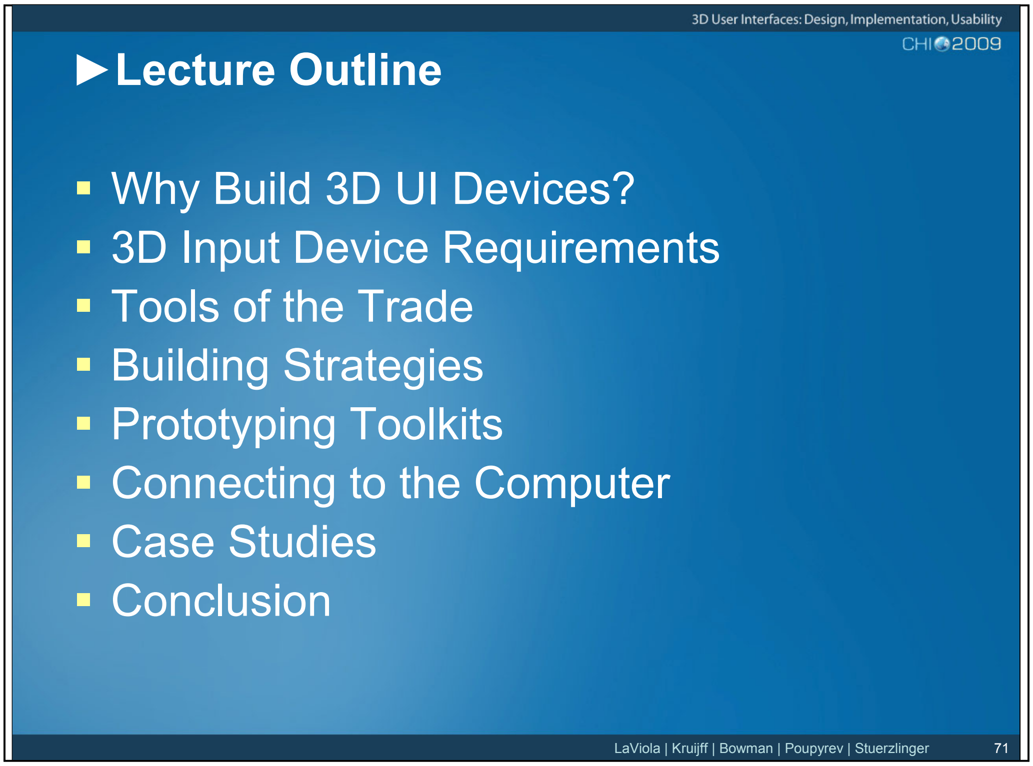
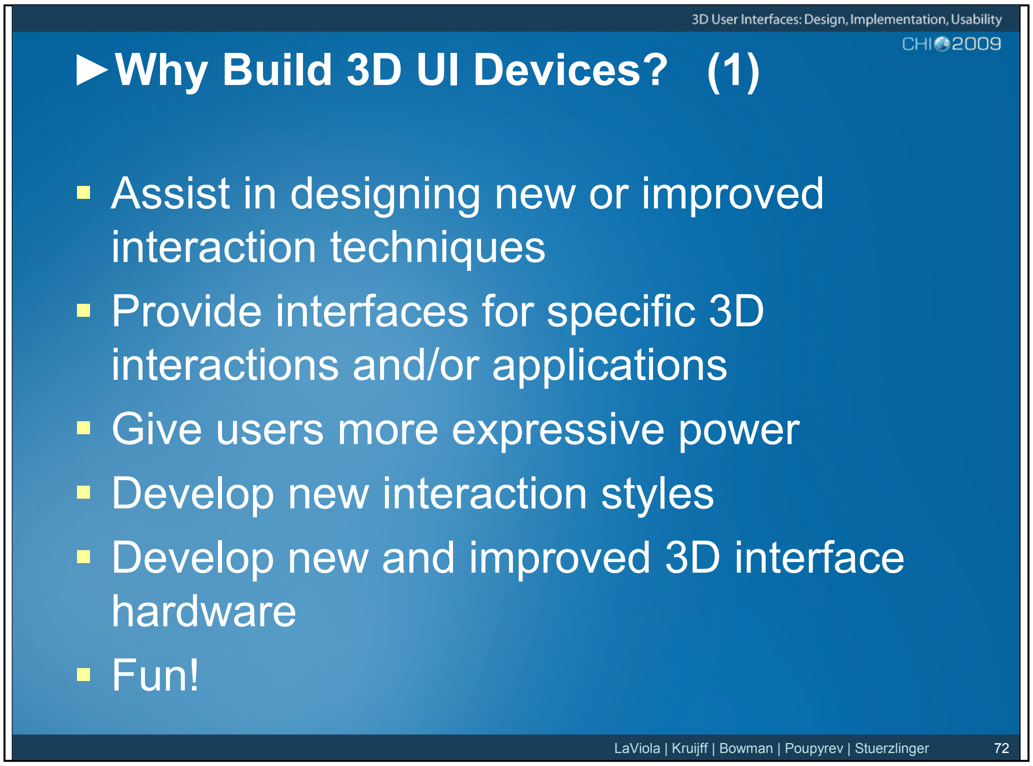
3D UIs 201

Ernst Kruijff



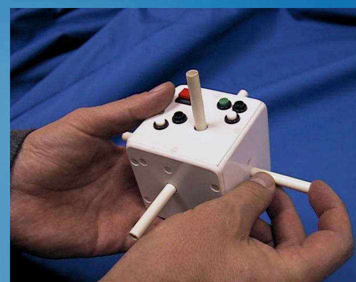
In this lecture, we will be talking about constructing custom built input device hardware. We will discuss why it is important to build custom-made 3D input devices. We will discuss the tools that you need to construct these devices and also some strategies for building the physical hardware and connecting them to a computer. We will examine several custom built devices through three case studies.

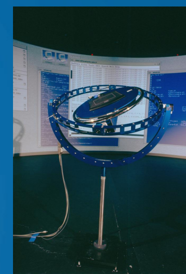


Why do would we ever want to build our own 3D input devices. Besides it being a fun and rewarding experience, there are several reasons to do so in the context of 3D user interface design. First, building new and interesting input devices can help user interface developers and researchers design novel interaction techniques and improve on existing techniques. There are many cases where a given interaction technique was designed using a particular input device and a new device would make significant improvements to the technique in user performance, ease of use, and ergonomics. Second, there are many instances in 3D user interfaces and 3D applications where that require specific forms of interaction and a device well suited to these forms may not be available. Thus, building a custom input device would greatly improve usability for that particular technique or application. Third, as a general rule, we always want to find new and innovative ways to interact with computers, especially in 3D, and custom built devices are way to explore that space.

►Why Build 3D UI Devices? (2)

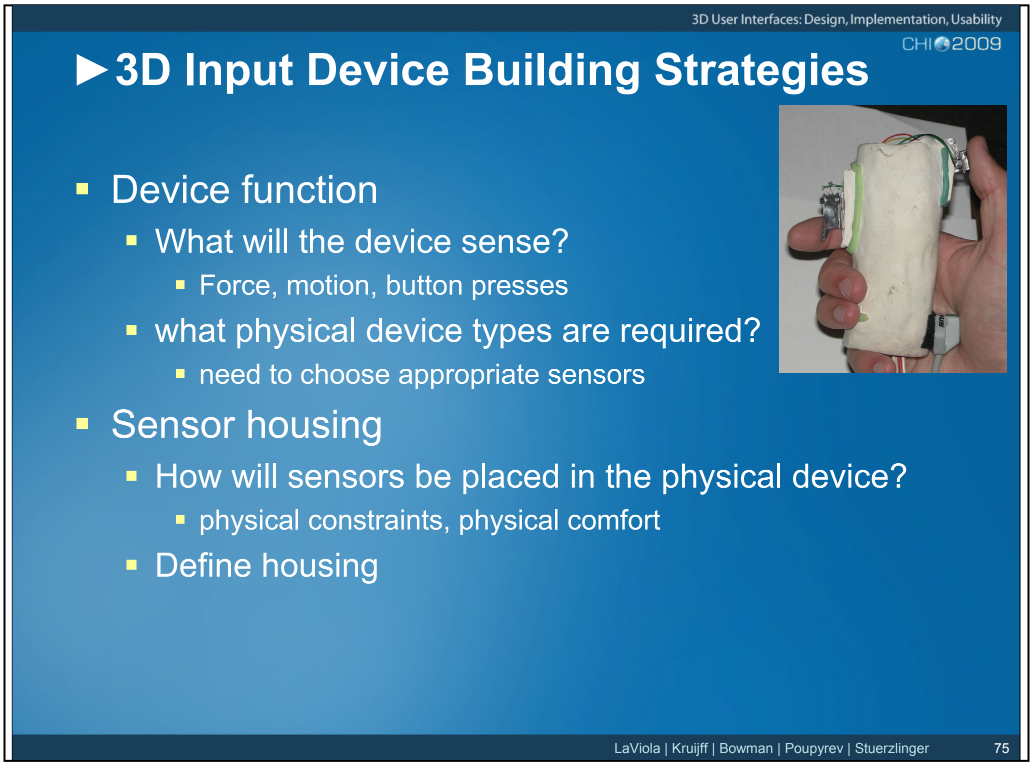
CubicMouse / variants
by Froehlich / Plate *et al*





CAT by Hachet *et al*

Building custom made input devices requires a variety of different components including microcontrollers, device controls such as buttons, sliders, switches, and various sensors. Also needed are cables to connect the device to the computer, power supplies, and some type of housing for the electronics and the device itself. A soldering iron is also a common tool to fuse connections and a circuit breadboard for prototyping electronics.



There are several strategies and guidelines to follow when building 3D input devices. Having a good plan of attack and answers to the following questions will make it much easier to build a device.

What will the device sense?

It is import to know what the device is supposed to do. Will it just have several buttons or will it sense motion, force, temperature, etc…

What physical device types are required?

Once you have an idea for what the device is doing to do, you need to determine how it is going to do it. For example, will the device need to convey digital information or analog or both. What kind of sensors will be required for the device to observe its surroundings? Will it need bend sensors, accelerometers, potentiometers, etc… A great sensing material is conductive cloth due to its flexibility and low cost. We will see examples of how conductive cloth is used in building custom devices later on in the lecture.

How will the sensors and buttons be placed in the physical device?

The device is going to require some type of housing and it is important to ensure that any controls the user actively must invoke are placed in or on the physical housing so that the user is comfortable and there is no undue physical strain.

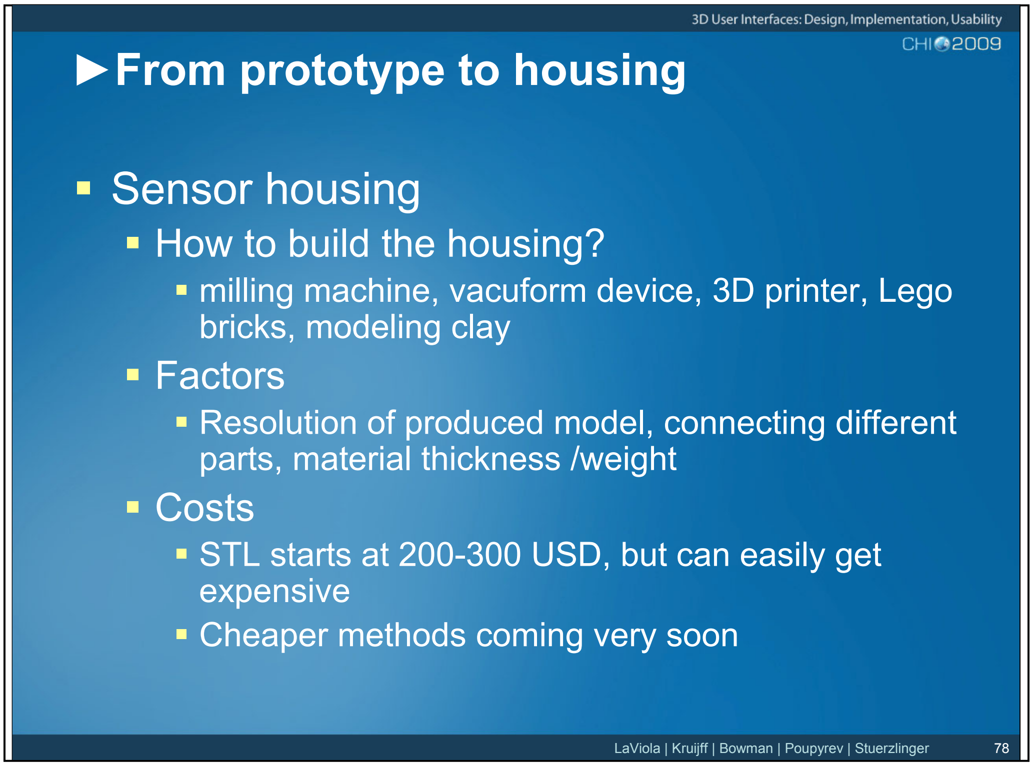
How to build the physical housing of the device?

There are many different approaches to building the device housing. A milling machine or vacuform device would probably do the best job (along with some 3D modeling software). Alas, not everyone has access to these machines. Less expensive alternative include modeling clay and Lego bricks.

Another important consideration when building a input device is ergonomics. Good ergonomic design is crucial when constructing input devices. The device should be lightweight so as to avoid fatigue and undue strain. It should also be simple to use and make it easy to reach all of the buttons and controls. The picture in the slide shows an input device called the CyberGrasp. This device provides force feedback on the users fingers. Unfortunately, the device has rather poor ergonomic design. However, this is more a function of the state of the art in haptic technology.

▶ Design ergonomics – how to?

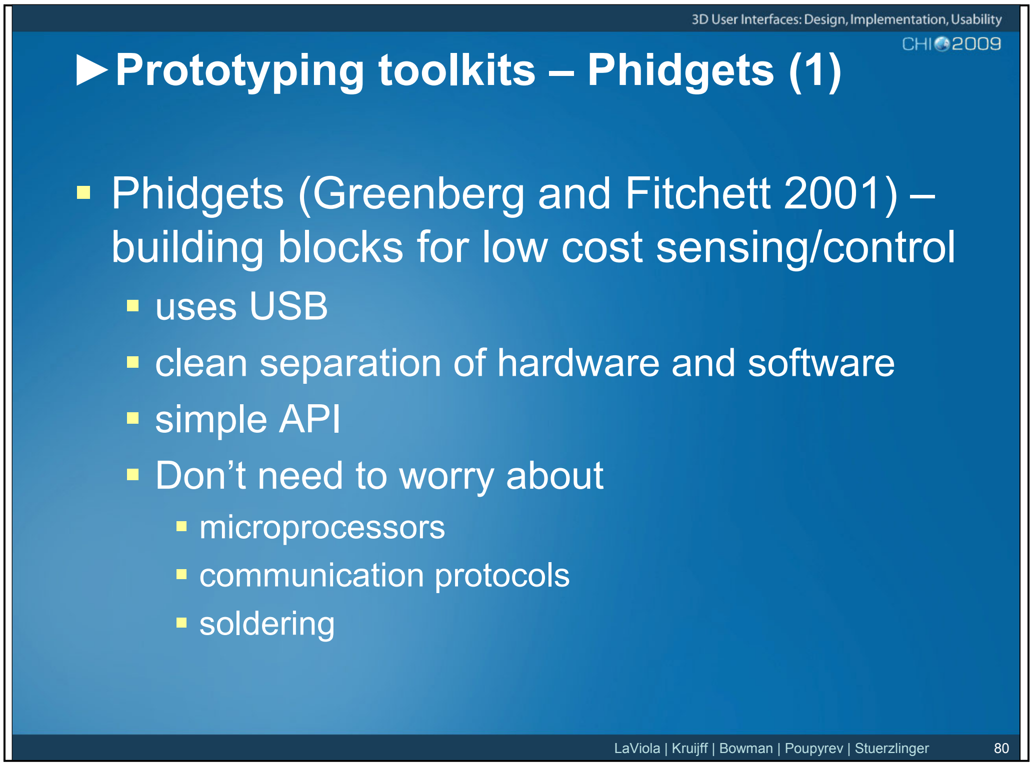
- Use cardboard boxes
- Build clay models
- Put dummy controllers on prototypes to check placement



There are several ways of producing a housing for an input device. Whereas for starters, some simple building material like clay can be used, for the final prototype it is recommended to make use of solid material like STL produced models. Nowadays, models are not so hard to create and new production methods are coming available that provide for low-cost production.

►Where to get your controllers

- Break apart your (old) devices
  - search the “mouse graveyard…” of your system administrator
- Buy controllers at regular stores (Radioshack, Conrad, etc.)
- Use a prototyping toolkit

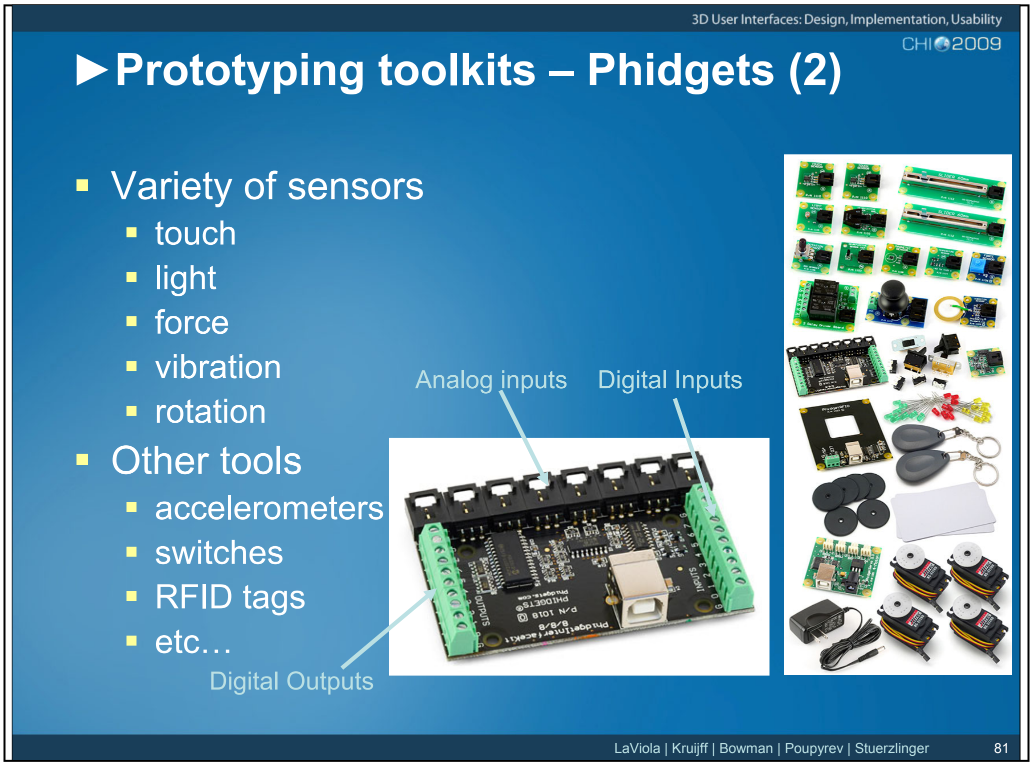


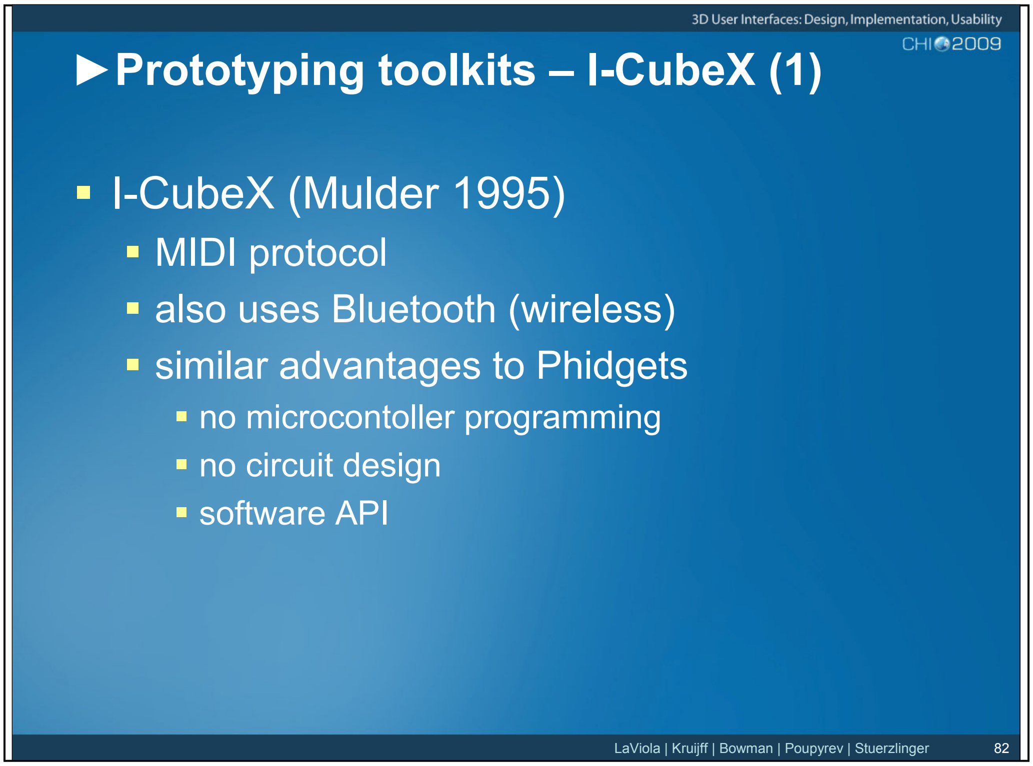
We have seen that there are many things to think about when designing and building a custom made input device. To make it easier to develop new input devices, there are a number of prototyping toolkits that mask unwanted details in the development process. Phidgets are one example of a set of building blocks for developing input devices. Phidgets provide a variety of sensors and other tools to make it easy to create input devices without having to worry about microcontrollers, communication protocols, device drivers, and soldering.

References:

www.phidgets.com

Greenberg, S. and Fitchett, C. Phidgets: Easy Development of Physical Interfaces through Physical Widgets. *Proceedings of the UIST 2001 14th Annual ACM Symposium on User Interface Software and Technology,* 209-218, 2001.


3D User Interfaces: Design, Implementation, Usability
CHI 2009
►Prototyping toolkits – Phidgets (2)
Variety of sensors
touch
light
force
vibration
rotation
Other tools
accelerometers
switches
RFID tags
etc…
Analog inputs
Digital Inputs
Digital Outputs
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
81

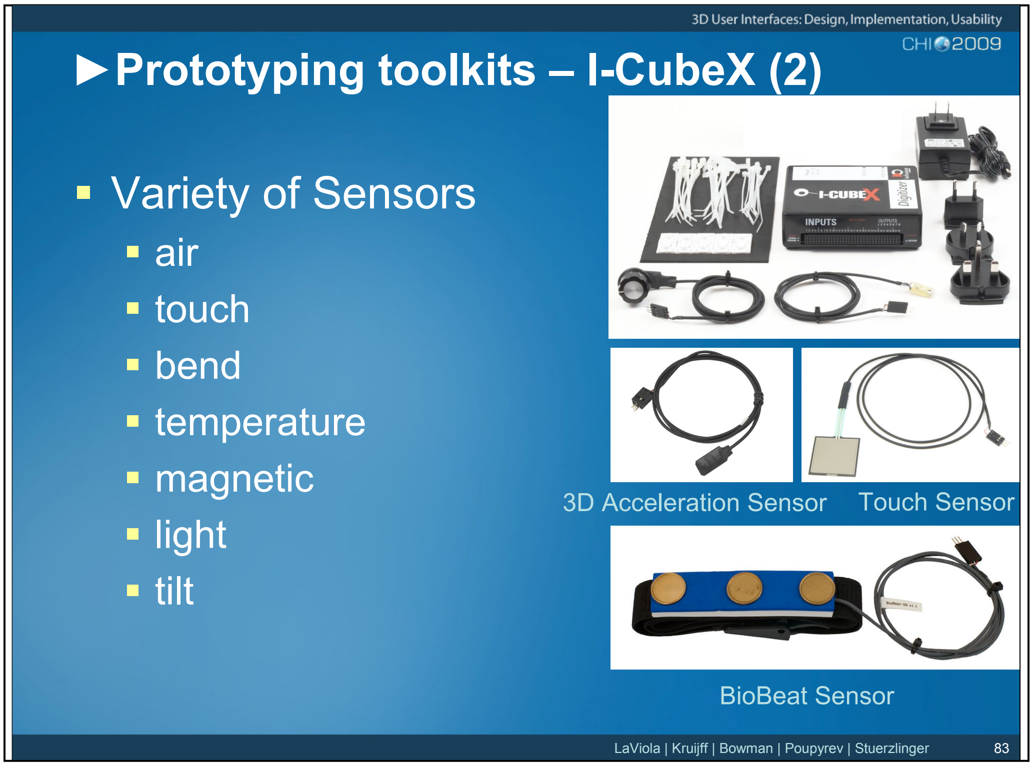


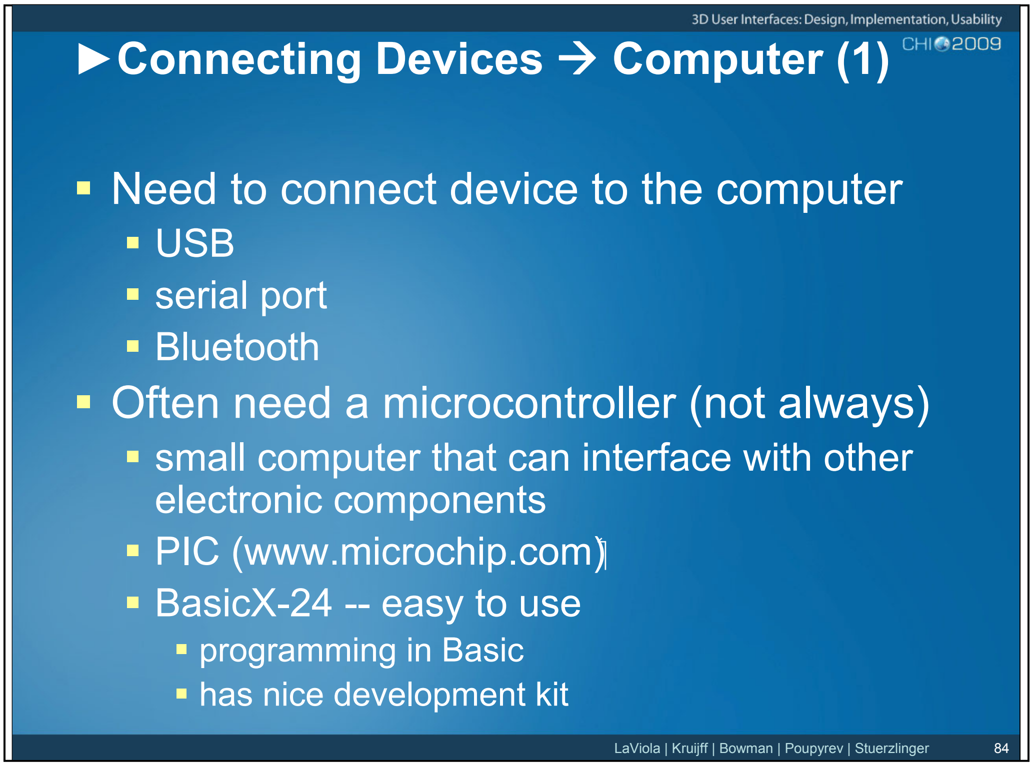
Another toolkit for prototyping and building new input devices is the I-CubeX system. I-CubeX makes use of the musical instrument device interface (MIDI). It also utilizes Bluetooth for wireless communication. The I-CubeX toolkit has similar advantages to Phidgets and a plethora of sensors comparable to Phidgets as well.

References:

infusionsystems.com

Mulder, Axel. The I-Cube System: Moving Toward Sensor Technology for Artists. *Proceedings of the 6th International Symposium on Electronic Art,* Montreal, QC, Canada, 1995.


3D User Interfaces: Design, Implementation, Usability
CHI 2009
►Prototyping toolkits – I-CubeX (2)
Variety of Sensors
air
touch
bend
temperature
magnetic
light
tilt
I-CUBEX
INPUTS
3D Acceleration Sensor
Touch Sensor
BioBeat Sensor
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
83

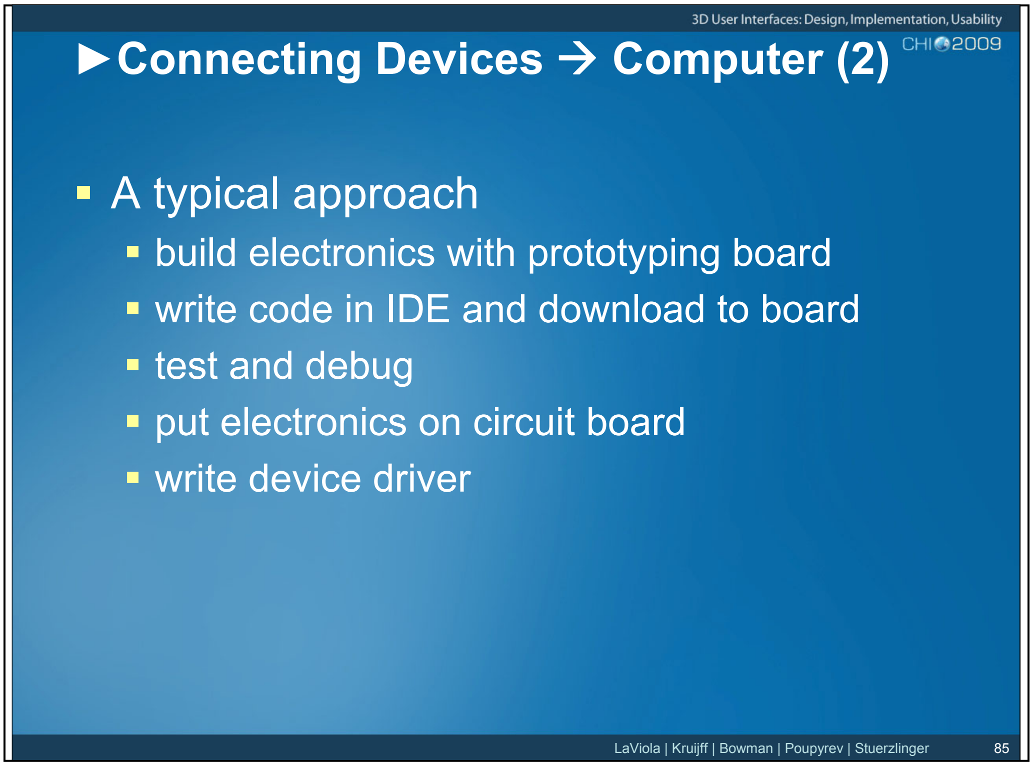


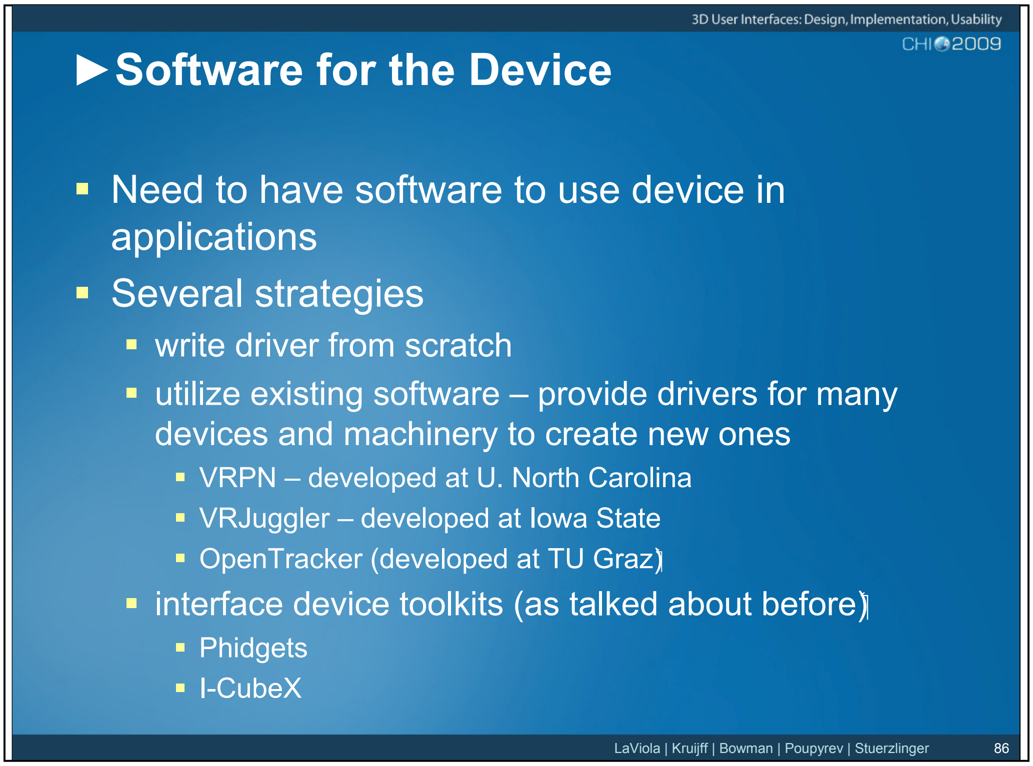
When building input devices, we must have as way to connect them to the computer. There are several different approaches to doing so including USB, serial port, and Bluetooth. Very often a microcontroller is needed as a mechanism for communicating to the computer from the physical device. A microcontrollers is simply a small computer that can interface with various electronic components. There is a wide variety of microcontrollers on the market today varying in size, functionality, and power consumption. Two of the most common are PIC and BasicX microcontrollers. They both have development kits and are programmed with either Basic or C.

Typically, building a device consists of building the electronics on a prototyping board, writing the necessary logic, and downloaded it to the board for testing and debugging. Once testing is done, the electronics could be put on a circuit board. A device driver then would need to be written so the computer could communicate with the device.

A great resource for getting started with microcontrollers is

Forman and Lawson, Building Physical Interfaces: Making Computer Graphics Interactive, Course #30, SIGGRAPH 2003.


3D User Interfaces: Design, Implementation, Usability
CHI 2009
Connecting Devices → Computer (2)
A typical approach
build electronics with prototyping board
write code in IDE and download to board
test and debug
put electronics on circuit board
write device driver
LaViola | Kruijff | Bowman | Poupyrev | Sturzlinger
85



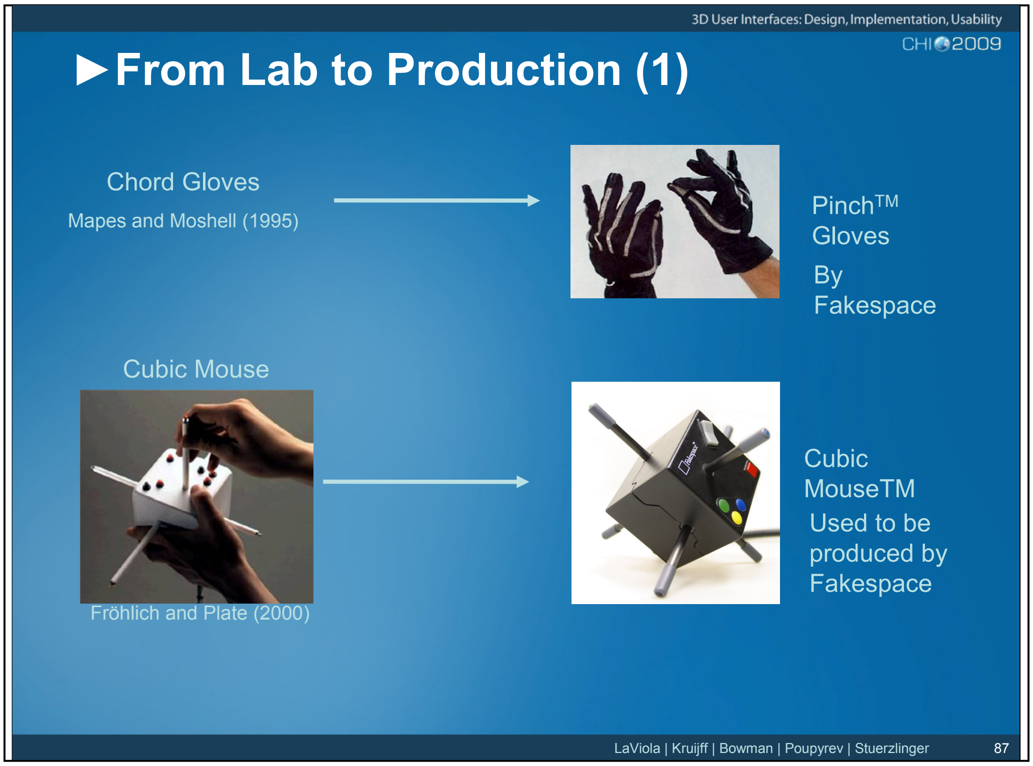
Once the device is built and along with the necessary electronics, software is needed to interface to it so that people can use the device in their application. One approach is to write the device driver from scratch. Typically one has to know something about the operating system as well as understanding the appropriate communication protocols (serial/USB/Bluetooth). Functions that are often required are open, close, read, and write. The device driver can then be incorporated into an API so developers and researchers can use it.

A better approach is to use existing software to assist in getting the device ready for use in applications. There are several software frameworks that make it easier to create device drivers. For example, VRPN, developed at the University of North Carolina, Chapel Hill, provides a framework for connecting devices to applications. It currently supports many devices and provides infrastructure to create new drivers. Another example is VR Juggler, developed at Iowa State.

Probably the best approach is to make use of interface device toolkits. We will look at two examples of them later in the lecture.

References:

http://www.cs.unc.edu/Research/vrpn/
http://www.vrjuggler.org/

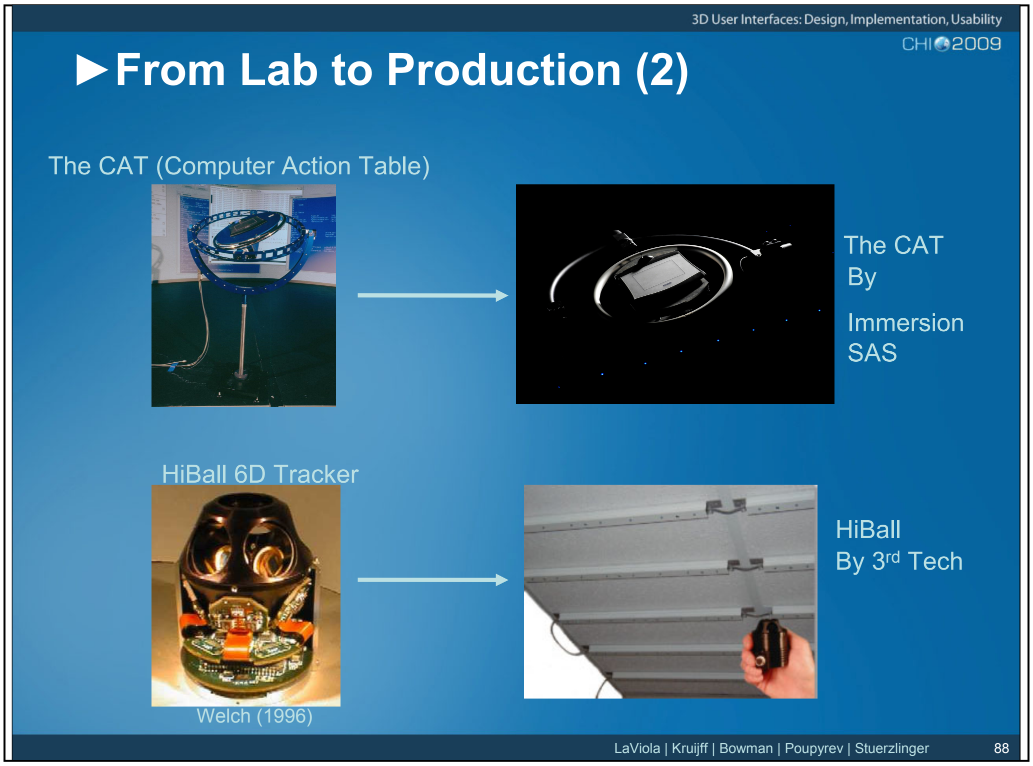


There have been a number of cases where custom built input devices from research laboratories have been successfully commercialized. The next two slides show some examples.

References:

Mapes, Daniel P. and Moshell, J. M. A Two-Handed Interface for Object Manipulation in Virtual Environments, *Presence: Teleoperators and Virtual Environments*. Vol. 4(4):403-416. Fall 1995.

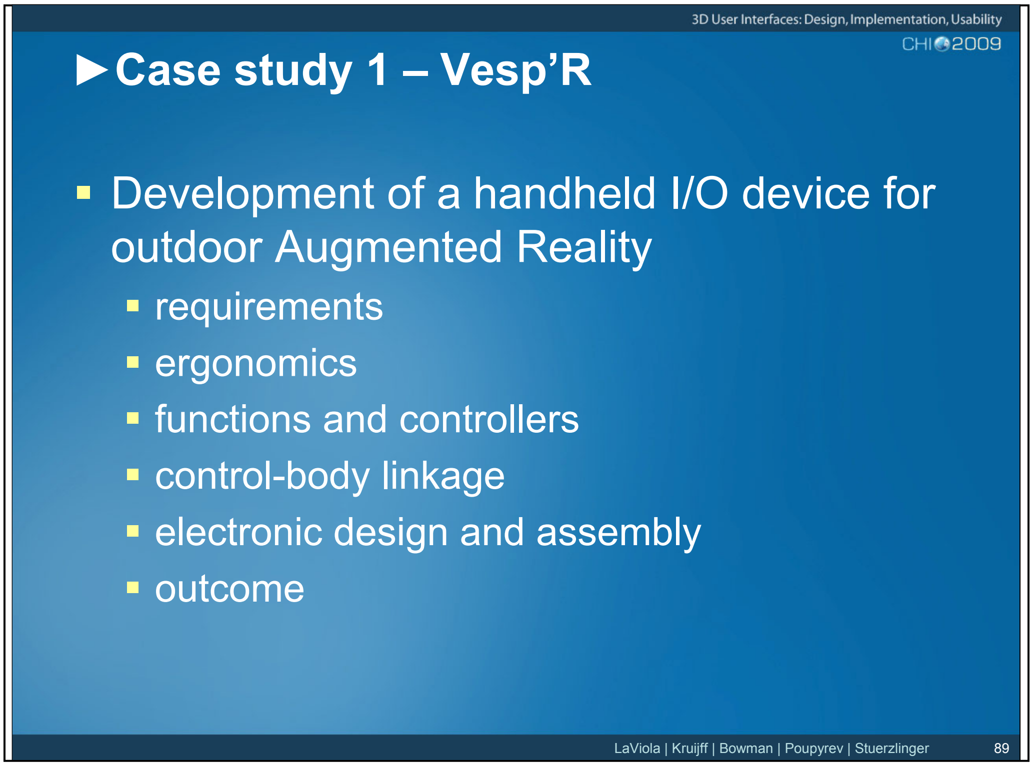
Fröhlich Bernd and John Plate. The Cubic Mouse: A New Device for Three-Dimensional Input. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), 526-531, 2000.



References:

Hachet, M., Guitton, P., and Reuter, P. The CAT for Efficient 2D and 3D Interaction As An Alternative to Mouse Adaptations. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (VRST '03), 225-112, 2003.

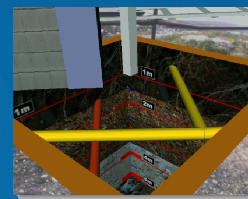
Welch, Greg, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci High-Performance Wide-Area Optical Tracking: The HiBall Tracking System, *Presence: Teleoperators and Virtual Environments* 10(1): 1-21, 2001.



More details on this case study can be found in:

Veas, E. and E. Kruijff. Vesp'R - design and evaluation of a handheld AR device. In Proceedings of the 7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'08). 2008

▶Vesp'R requirement analysis



vidente
Schall et al / TU Graz

Application requirements?

- outdoor AR project (Vidente)
  - overlay underground structures for field workers
  - lenses, visualization filtering, labeling, snapshot
  - higher complexity application
  - extended device construction required (sensors, controllers)
  - traditional plastic case solution insufficient

►Vesp'R requirement analysis (2)

Needs

- integrate external devices without destroying ergonomics
- improve grip
- add accessible controllers
- keep weight limited and better balanced

▶Vesp'R ergonomic analysis

Which ergonomic factors?

- analyse SONY Vaio UMPC (popular platform)
- device held up rather high causes fatigue
- two-handed power-grip, single-handed grip limitations
- wrist problems associated with control access, fine actions problem

►Vesp'R ergonomic analysis

Outcomes

- better pose can aid in lowering fatigue
- separation of grip („handle") and display relieve wrists
- bigger and better placed controllers needed
- direct trade-off between pose, weight, and task performance in longer duration sessions

▶Vesp'R functional allocation

How to map functions to controllers?

- Vidente functionality can be performed with limited number of controllers
  - 1D buttons for system control / visualization mode changes,
    2D controllers (joysticks) for other functions
  - take into account thumb / index finger control
    - fine motor control, but may lead to disbalance
  - no spatial control (outside cam), no pen control

▶ Vesp'R control-body linkage

What grip?

- Handles resemble the usual form (drill, joystick)
  - ideal power grip: 76mm from fingers to palm

Where to put the controllers?

- using off-the-shelf controllers and wireless MIDI interfaces
  - space for controller board, cabling, batteries

►Vesp'R control-body linkage

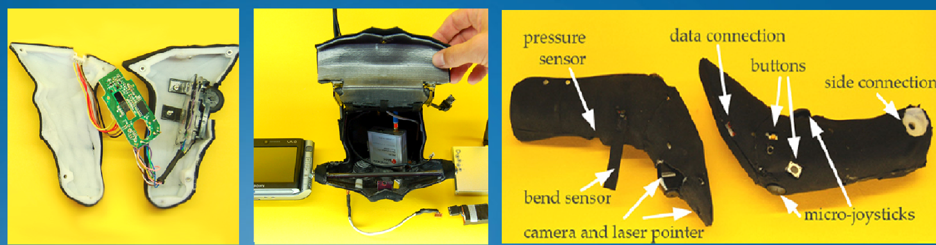
How to configure case and handles?

- mock-ups with foam and „abstract" handles
- distance between case and handles for grip
- placement of sensors (like GPS) is restricted
- check balance / weight distribution

Outcomes

- peripherals mounted behind UMPC, resulting in L-shaped base
- handles best placed at horizontal weight equilibrium removable handles

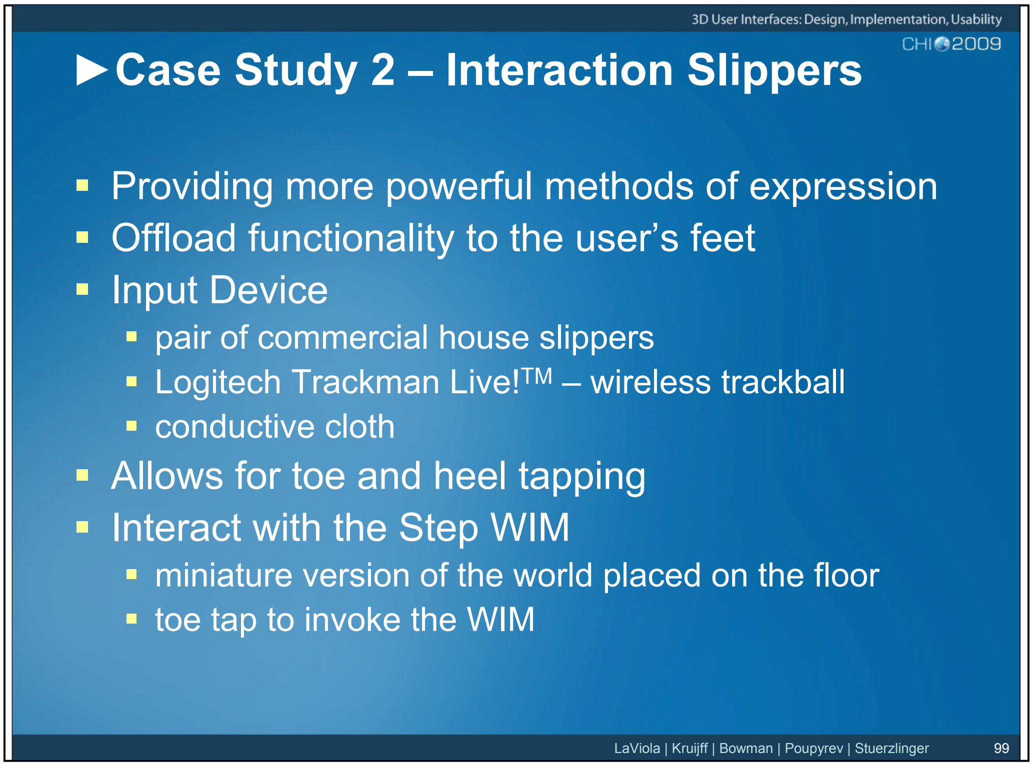
►Vesp'R electronic design / assembly



Where do the electronics go?

- detachable handles
- different USB and MIDI controllers, small controller boards in handles, adapted USB cables
- smaller handle with better power grip
- CAD models printed in STL, layered with velvety rubber

►Vesp'R outcome



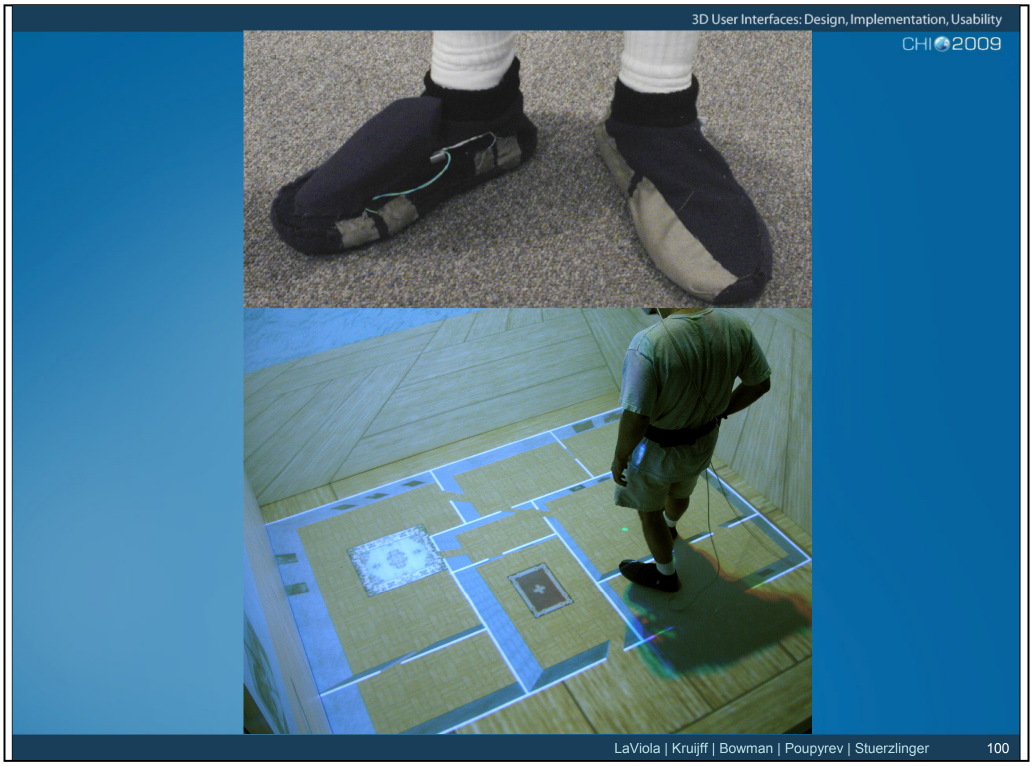
In our second case study, we look at the Interaction Slippers, a device for interaction in CAVEs, specifically the Step WIM.

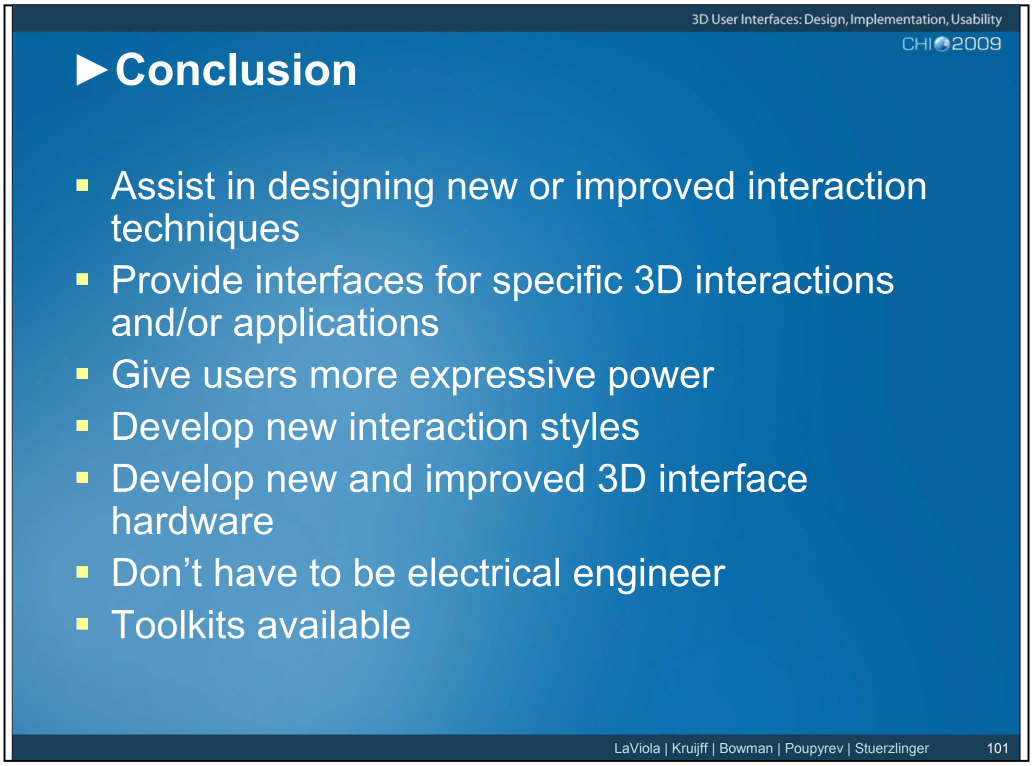
The Step WIM is a interaction widget for quickly navigating through a virtual environment. It is a miniature version of the world placed underneath the user's feet and acts as an augmented roadmap. The user can either walk around the Step WIM to get a better understanding of the virtual world or navigate to a specific place by simply walking to a desired location in the WIM and invoking a scaling command, causing the Step WIM to animate, scaling up around the user's feet, thereby seamlessly transporting the user to the specified location.

In order to invoke, navigate and dismiss the Step WIM, users wear a pair of slippers (slippers with an imbedded wireless mouse) which gives them the ability to perform toe and heel tapping. To invoke the Step WIM, users simply tap their toes together. The device uses conductive cloth so the buttons fit easily onto the slippers. A Logitech Trackman Live is imbedded into a pouch on top of the right slipper. The beauty of this design is that no special device driver is needed. Developers can simply use mouse button events.

Reference:

LaViola, J., Zeleznik, R., Acevedo, D., and Keefe, D. "Hands-Free Multi-Scale Navigation in Virtual Environments", *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, 9-15, March 2001.


3D User Interfaces: Design, Implementation, Usability
CHI 2009
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
100


3D User Interfaces: Design, Implementation, Usability
CHI 2009
▶Conclusion
Assist in designing new or improved interaction techniques
Provide interfaces for specific 3D interactions and/or applications
Give users more expressive power
Develop new interaction styles
Develop new and improved 3D interface hardware
Don’t have to be electrical engineer
Toolkits available
LaViola | Kruijff | Bowman | Poupyrev | Stuerzlinger
101