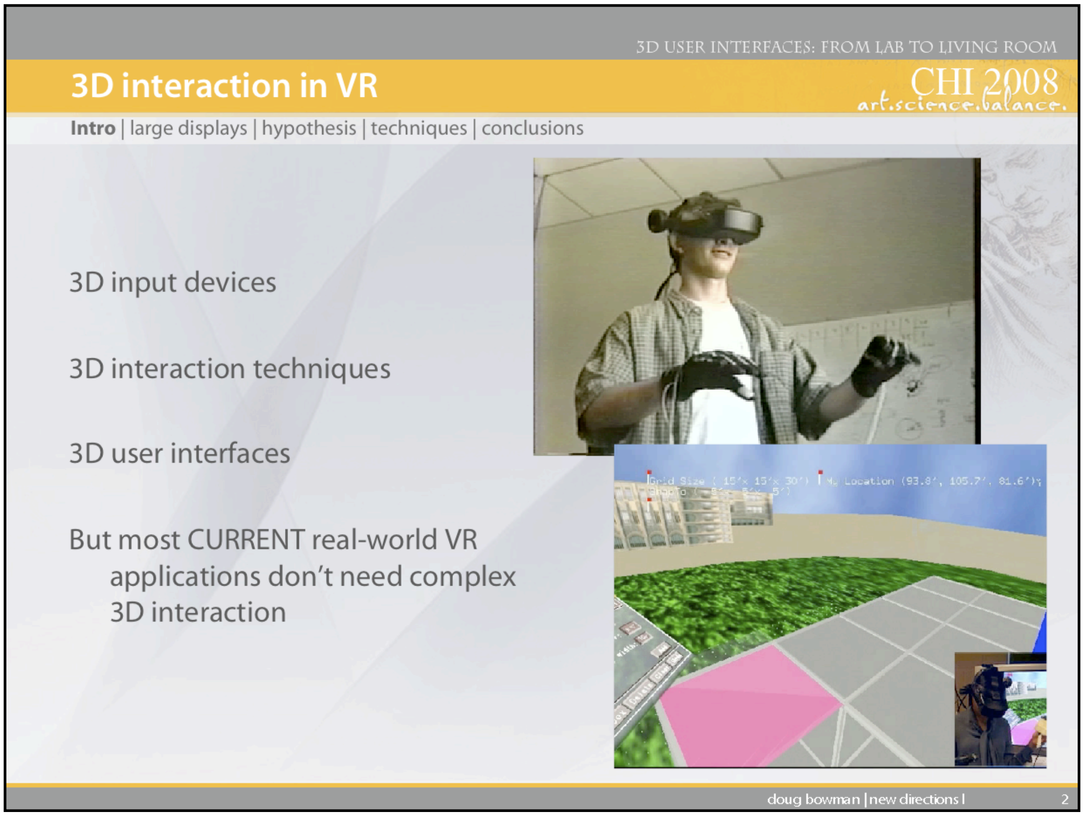
CHI 2008
art.science.balance.

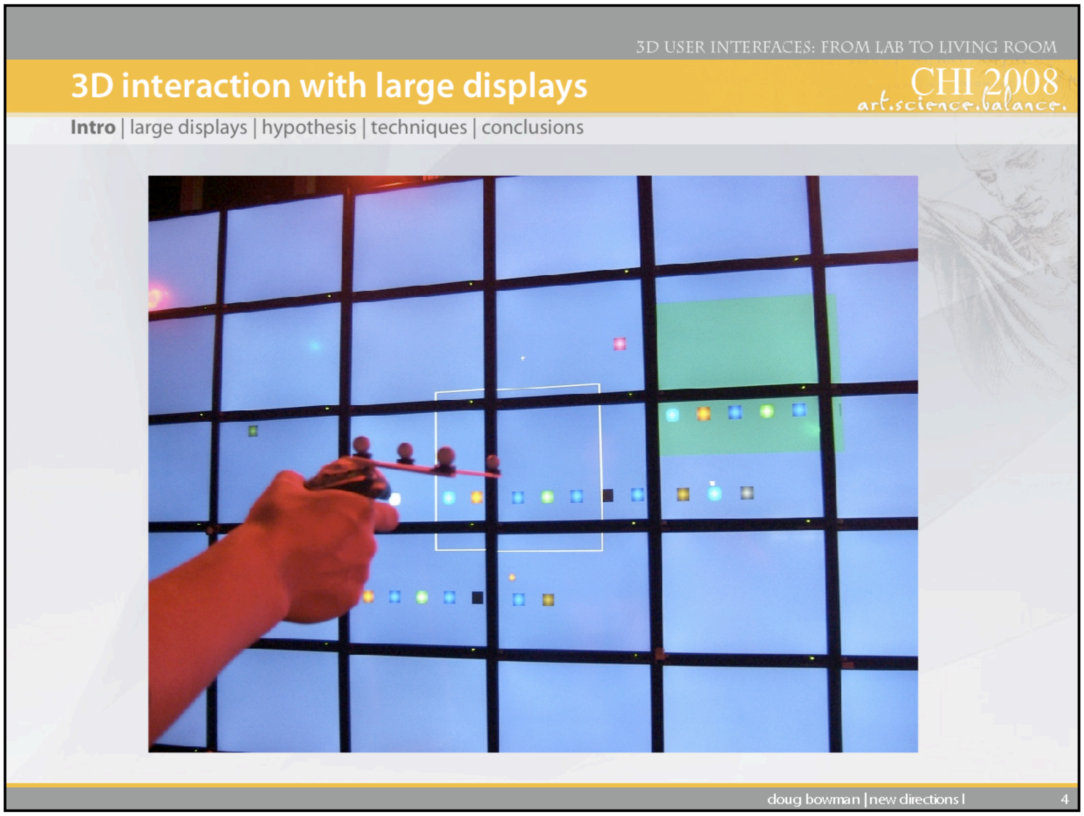
New Directions I: 3D interaction with large displays

Doug Bowman

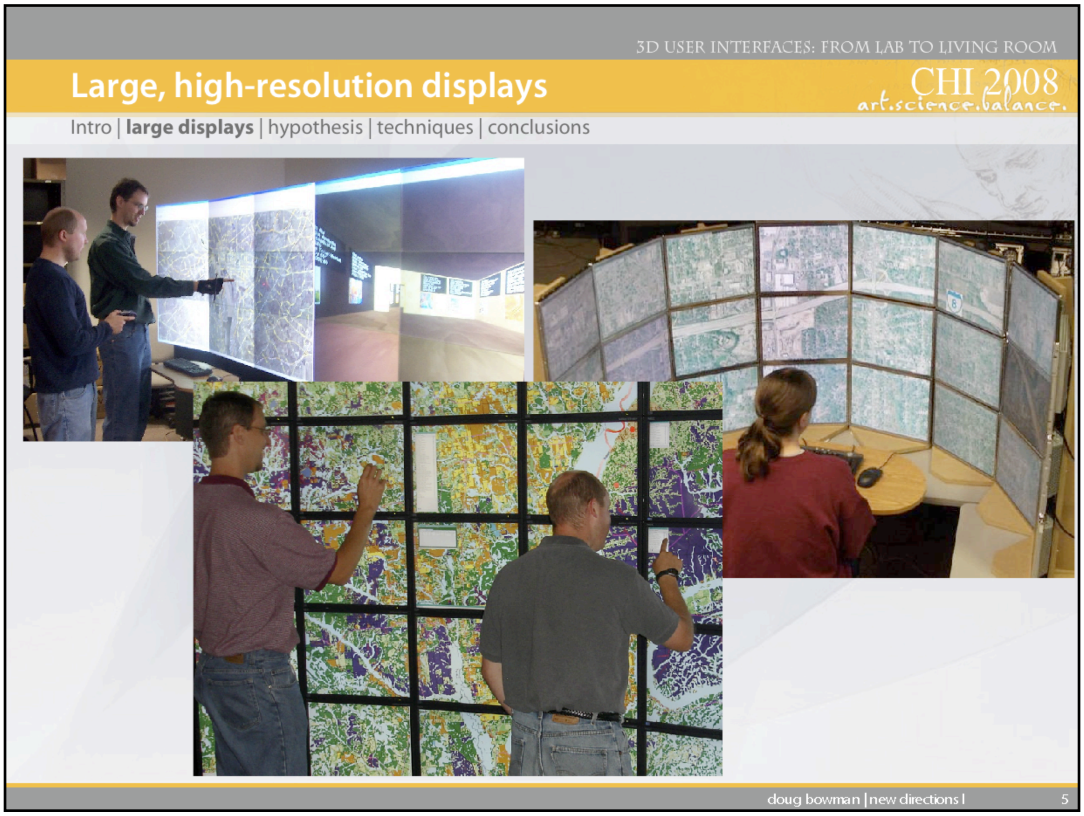


I've been doing research in this area for more than 10 years, and I think we've been quite successful in designing and evaluating some very usable 3D interfaces. But the problem is that almost none of this work has made it out into real-world VR applications. The work is good, but it's had very little impact outside the research community.

So what are some ways that the 3D interaction knowledge and techniques, which we developed in a VR context, can be applied outside VR?



We’re going to focus in this talk on the last example - 3D interaction with large displays.

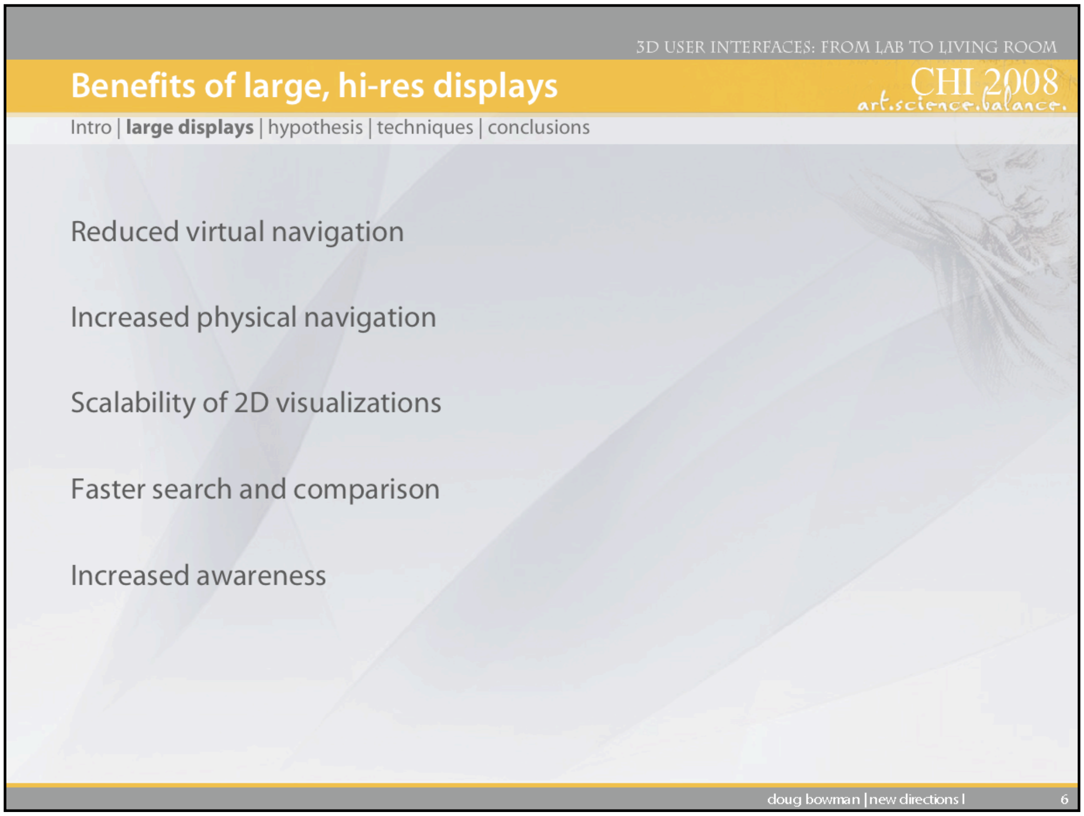


The context for this idea is the current trend towards large, high-resolution tiled displays. Here are three prototypes we are using in our lab.

Upper left: Rear-projected VisBlocks from VisBox, Inc. Each block has its own projector, and there are only minimal seams between the tiles. Moreover, each block is independent and can be moved to create new configurations of the display.

Upper right: Tiled LCD panels in reconfigurable columns of three displays. These can easily be curved around a single user, as shows in the picture, or flattened for multiple users.

Bottom: A larger tiled LCD array. This one has 50 panels in reconfigurable columns of 5 displays each. The entire display has 100 million pixels.

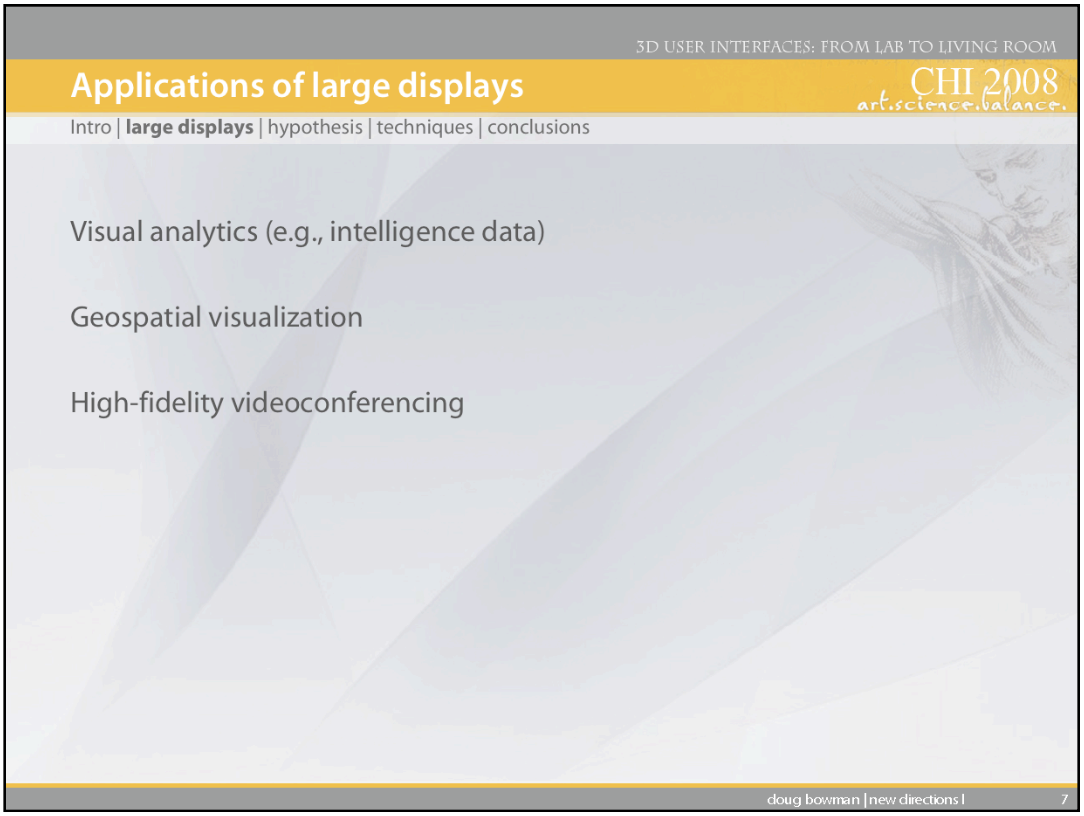


Empirical work has shown that displays of this type can deliver on their promise of better visual examination and analysis of data.

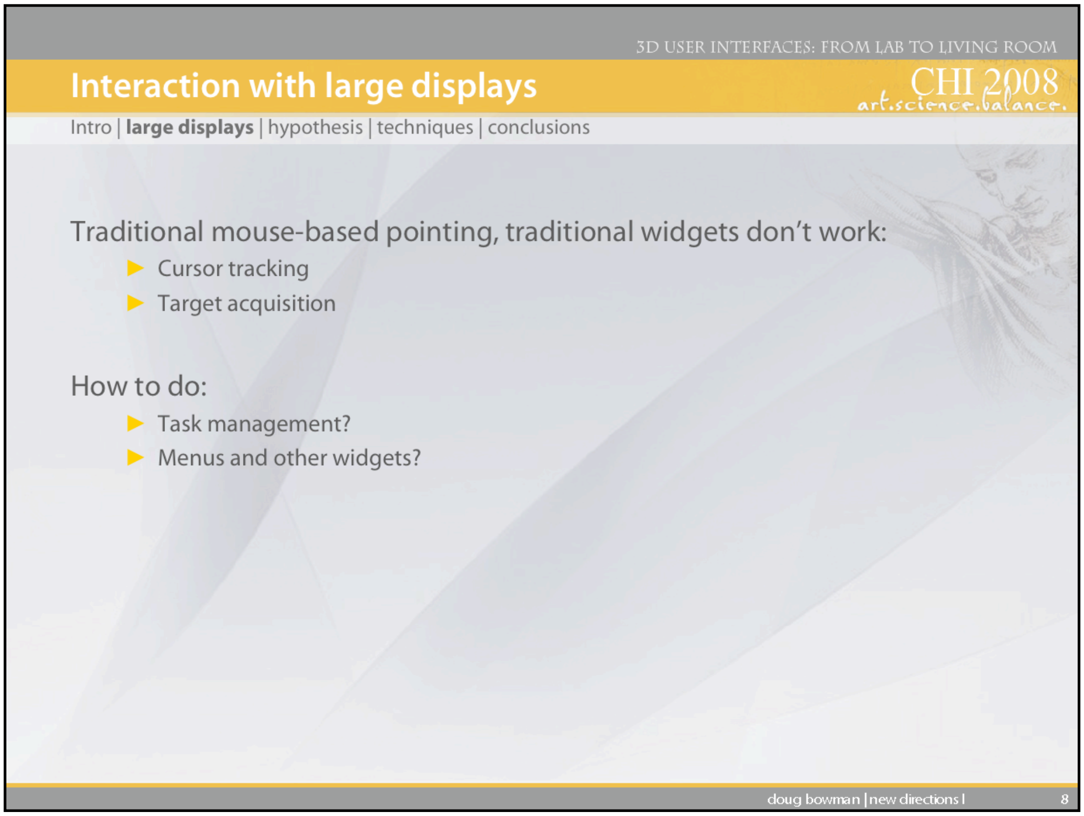
Ball, R. and North, C., Effects of Tiled High-Resolution Display on Basic Visualization and Navigation Tasks. in *ACM Conference on Human Factors in Computing Systems (CHI)*, (Portland, OR, 2005), ACM Press, 1196-1199.

Ball, R., North, C. and Bowman, D., Move to Improve: Promoting Physical Navigation to Improve User Performance with Large Displays. in *SIGCHI Conference on Human Factors in Computing Systems (CHI)*, (2006), 191-200.

Ni, T., Bowman, D. and Chen, J., Increased Display Size and Resolution Improve Task Performance in Information-Rich Virtual Environments. in *Graphics Interface*, (2006), 139-146.



Because of these benefits, such displays can be used for some high-impact applications.



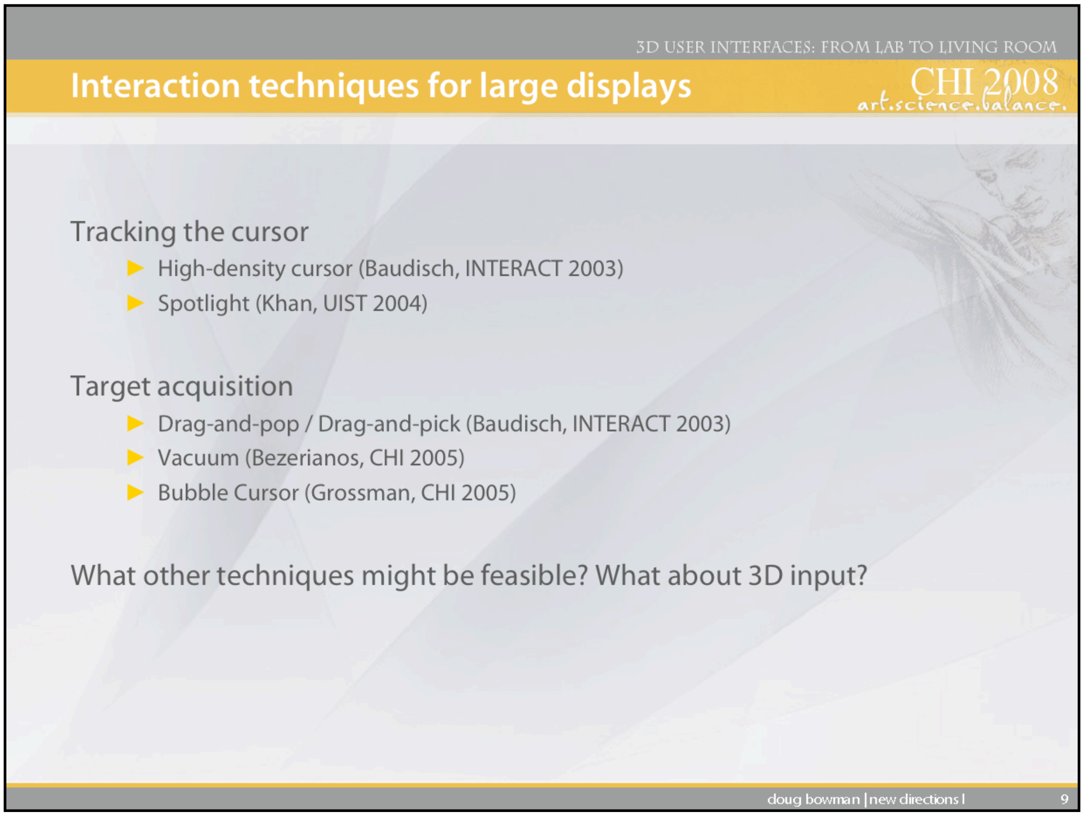
But it's also become very clear that traditional WIMP interfaces don't work very well on large, high-resolution displays.

Cursor tracking: the user loses the small cursor on the large display and has to search for it

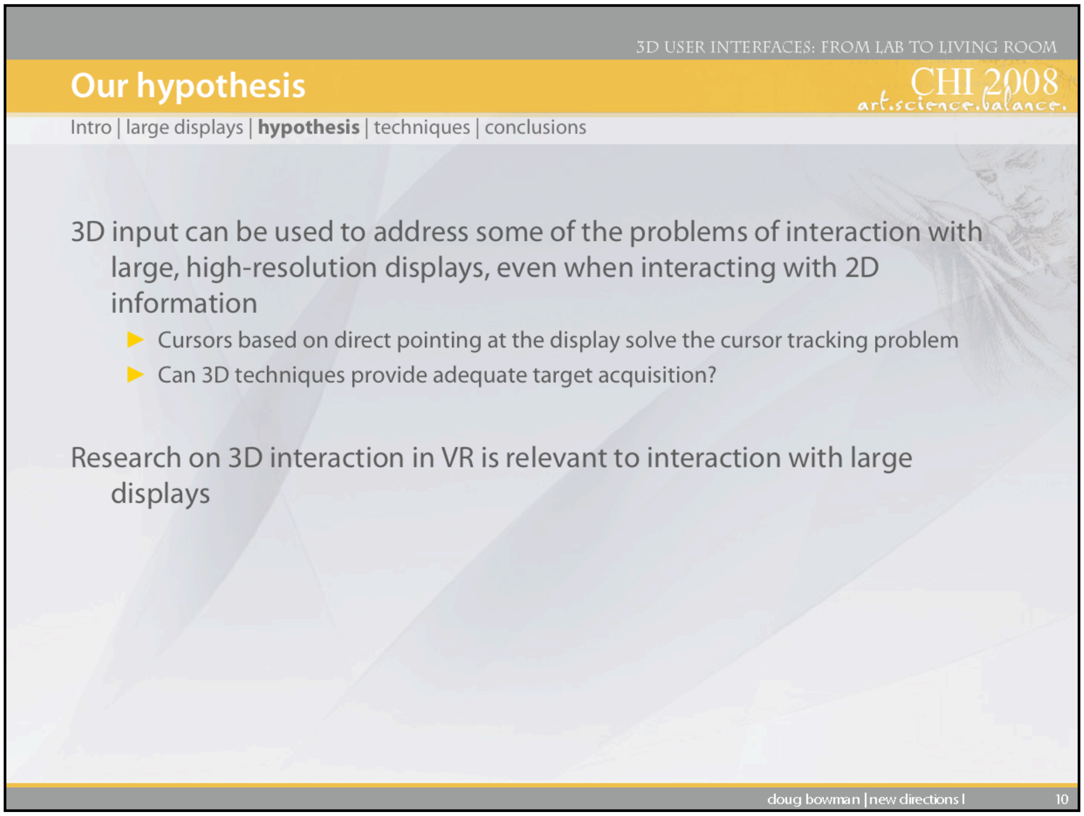
Target acquisition: it's difficult to select small objects because the mouse acceleration is either very high, or the user has to do a lot of clutching to move far across the display

Task management: these displays support a huge number of parallel windows/tasks

Menus/widgets: traditional menus and widgets don't work because they may require moving the cursor a great distance across the display, they may be too small to select or read from a distance, etc.

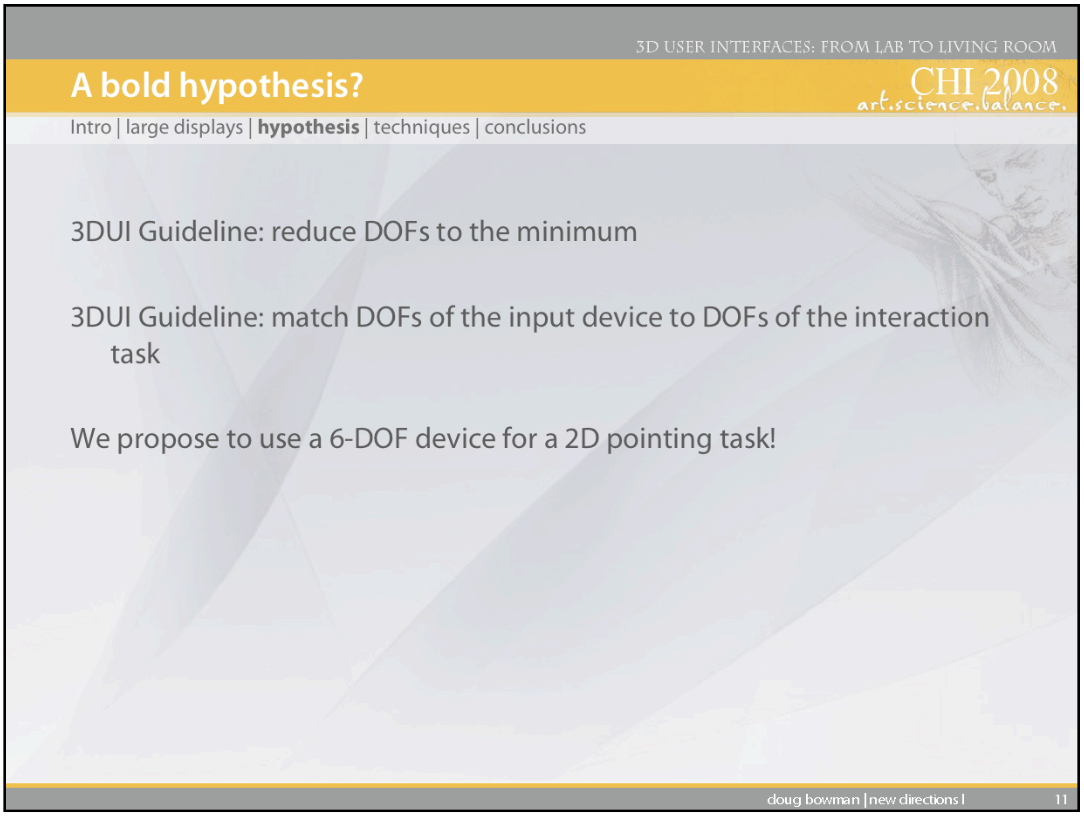


Because of these limitations, many researchers have developed enhanced mouse-based techniques for large, high-resolution displays. But these techniques still suffer from a fundamental problem: they tether the user to a fixed position (surface for the mouse). What if the user wants to walk around in front of the display?

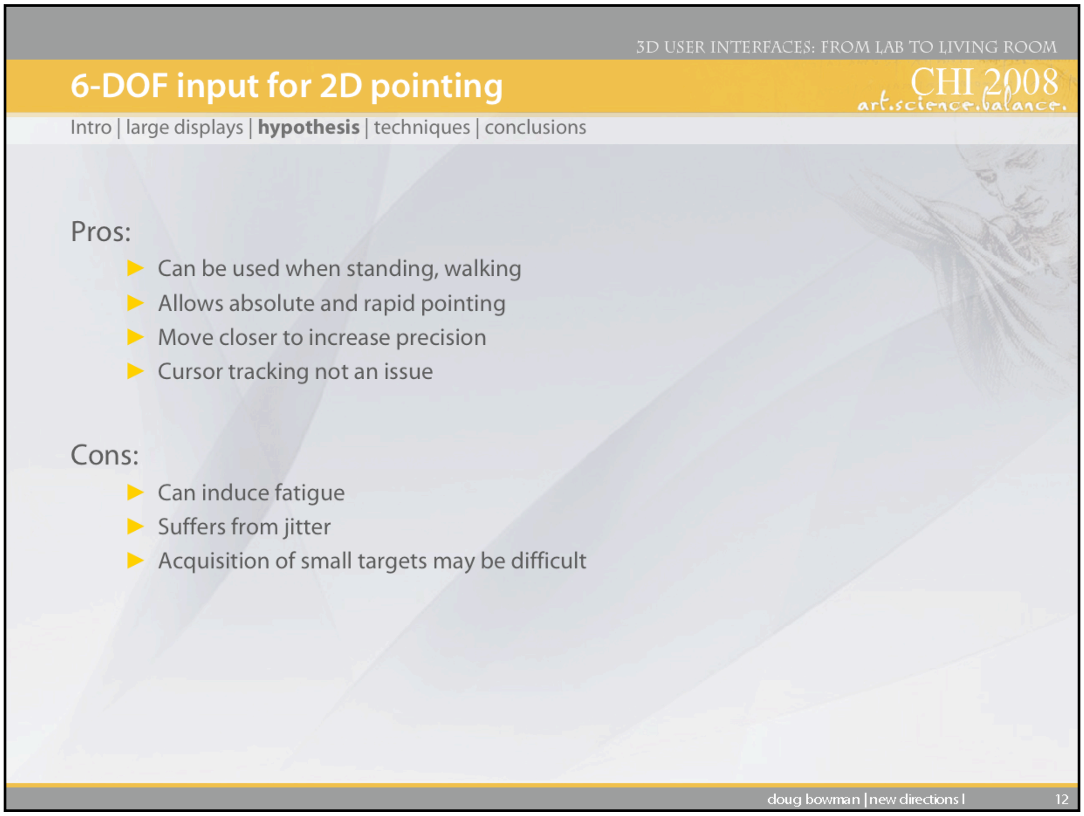


We hypothesize that 3D input (based on 6-DOF trackers) can solve some of these problems.

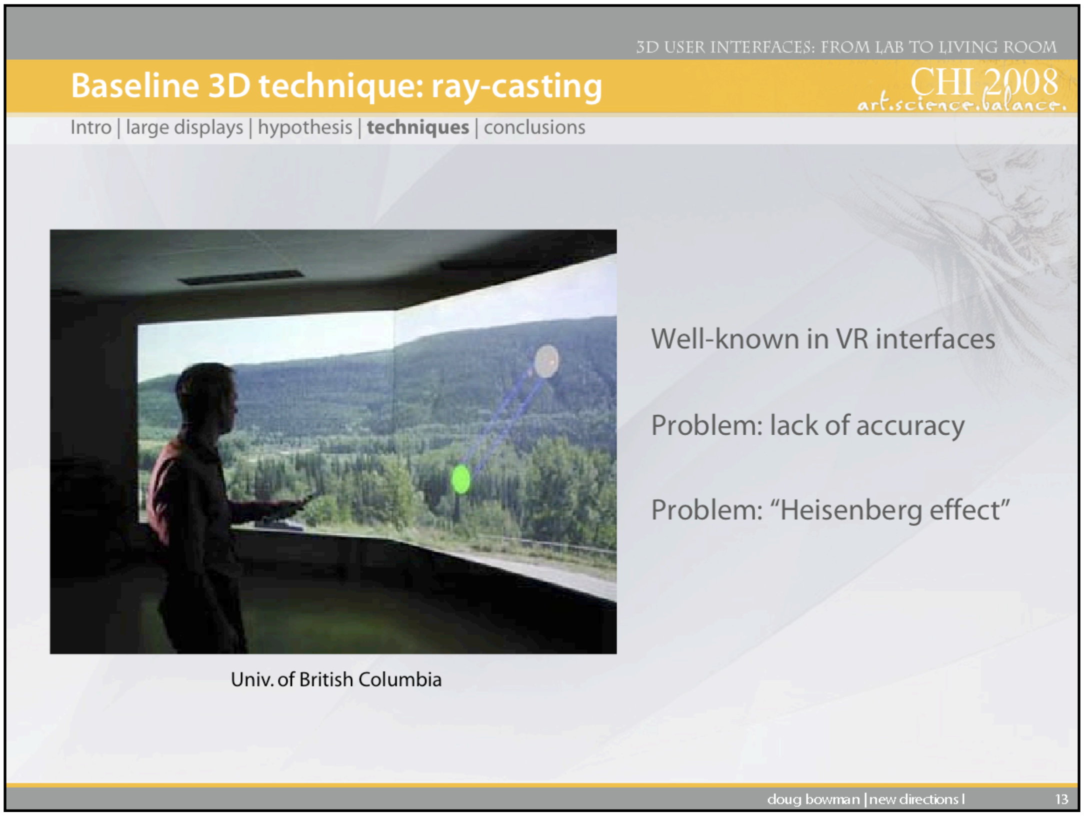
On a higher level, our hypothesis is that our work on 3D interaction in the context of VR is relevant to another (higher-impact?) area. That we can generalize our prior work and apply it to another real-world problem.



We've already seen (in the 3D UIs in a nutshell lecture) some guidelines that indicate this might not be a great idea!



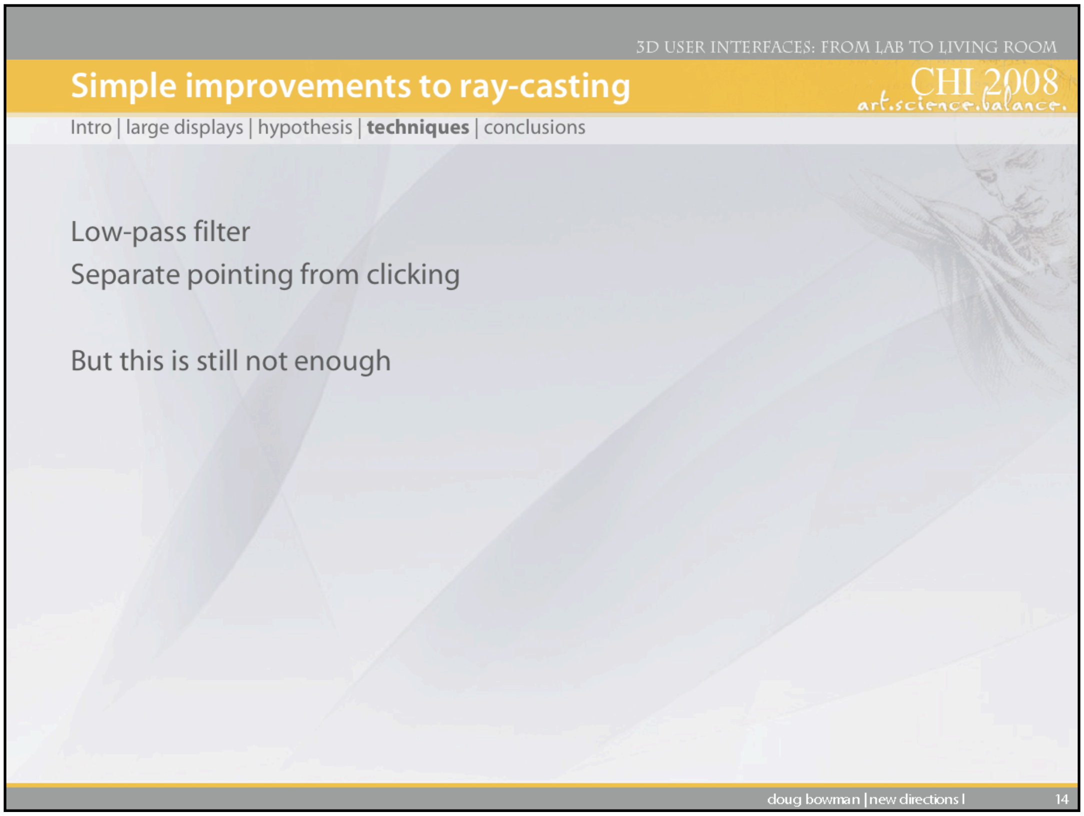
But there are some good reasons to use 6-DOF input for 2D pointing, despite the obvious disadvantages.



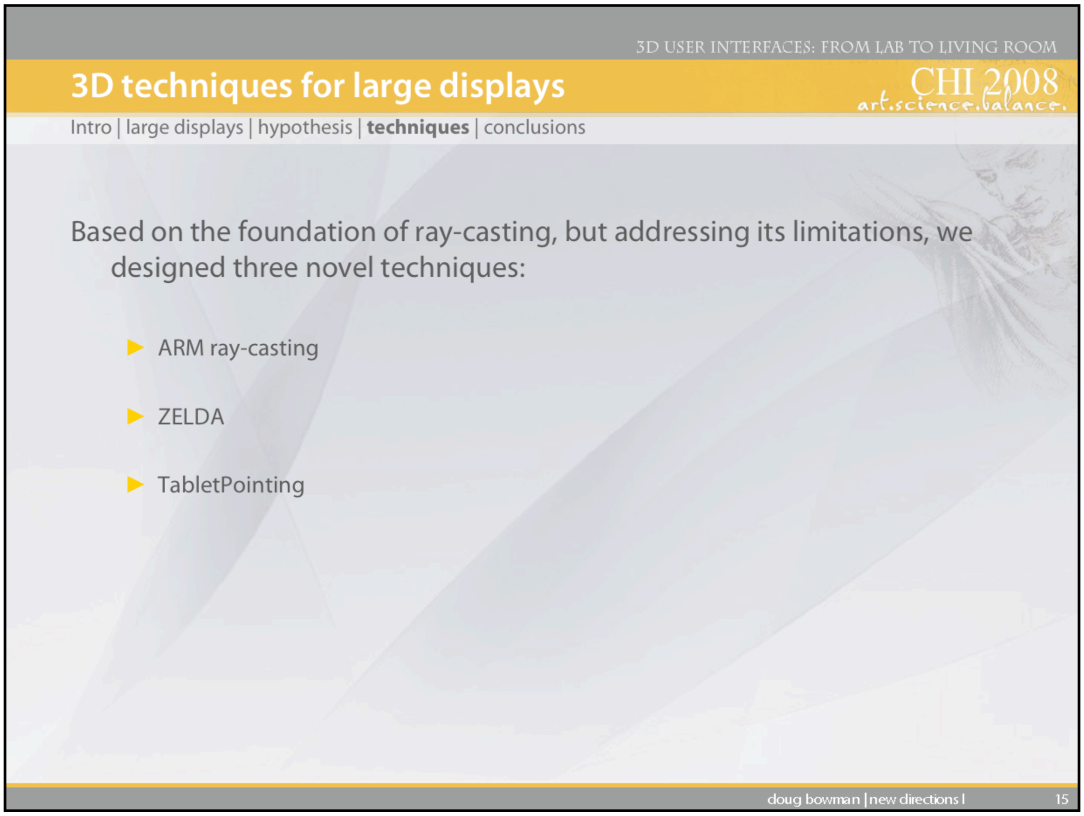
The obvious 3D interaction technique to start with is ray-casting. This simply means pointing the input device directly at the object/location you want to select/indicate. On a large display, the “laser pointer” metaphor easily describes this technique.

The problem, as anyone who has seen someone use a laser pointer from far away on a large screen will tell you, is that ray-casting from a great distance is not very accurate, because of hand jitter and because a small movement of the hand results in a large movement of the pointer.

Another problem is that pressing a button to indicate a selection can actually cause the device to move. This has been called the “Heisenberg effect” because it’s similar to the idea in physics that one can’t observe something without changing it.



We can easily improve ray-casting by applying a filter to reduce the noisiness of the data, and by pointing with one device/hand and clicking with another. But in our experience, this is not enough to make ray-casting usable on large, high-resolution displays.



So we set out to design techniques based on ray-casting, but addressing the accuracy issue.

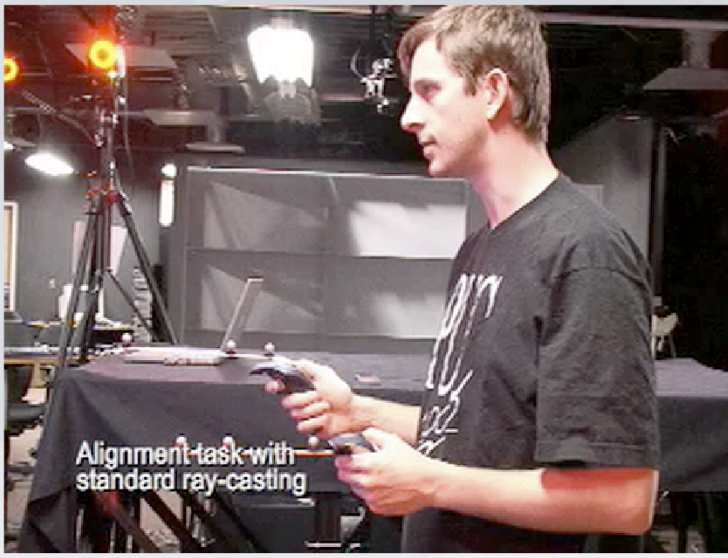
The idea of ARM Ray-casting is to allow the user to switch between absolute (standard ray-casting) and relative mappings between the pointing direction and the cursor position. The absolute mapping affords fast, coarse-grained interaction, while the relative mapping provides higher levels of precision. When relative mapping is activated, the current ray-casting intersection point is saved as the relative mapping origin. Any further intersections are then processed as vectors from this origin to prevent the cursor from jumping when relative mapping is activated. By scaling down these vectors by a constant factor, we effectively map standard ray-casting into a smaller defined area of interaction. With our application, we used a scale factor of 10 percent, which reduced the normal area of interaction (16000 pixels by 6000 pixels) to a small area of interaction (1600 pixels by 600 pixels). When relative mapping is deactivated, the cursor jumps back to the correct position for absolute mapping.

ZELDA uses a zoom window controlled by ray-casting to provide zooming capabilities. The zoom window displays underlying content – determined by the size of the window – in greater detail, which is based on the zoom factor. For example, with a zoom factor of 5 and a zoom window 1000 pixels by 1000 pixels, an underlying area of 200 pixels by 200 pixels, co-centered with the zoom window, is displayed five times larger than normal within the window. The remaining underlying area is occluded by the zoom window.

With ZELDA, we opted to use a bimanual interface for performing tasks. Many researchers have demonstrated that using the non-dominant hand for coarse tasks can be effective. Therefore, we chose to use the non-dominant hand to control the positioning of the zoom window and the manipulation of its size and zoom factor. We use the dominant hand to control the application cursor and to perform mouse-like interactions.

The scroll wheel on the non-dominant hand's input device is used to zoom in/out or resize the zoom window, depending on the device's orientation relative to the display.

ZELDA video



The TabletPointing technique takes a different approach to improving accuracy of ray-casting. It uses a tracked tabletPC as the input device, and rather than using it to point directly to targets on the large display, it is used to point to an interaction context on the large display. Once the desired context is highlighted, the user can use 2D interaction with the stylus on the tablet to select and manipulate targets.

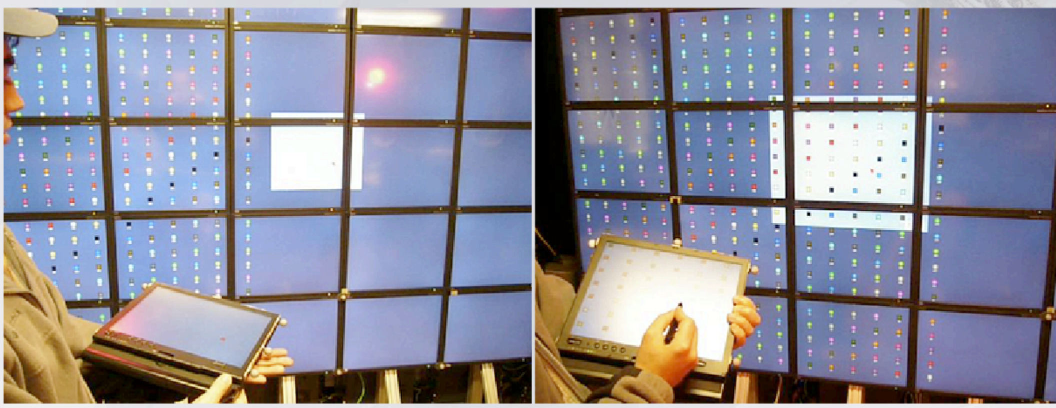
The user can change the size of the interaction context simply by walking closer (smaller context for more precise work) or farther away (larger context for coarser-grained tasks).

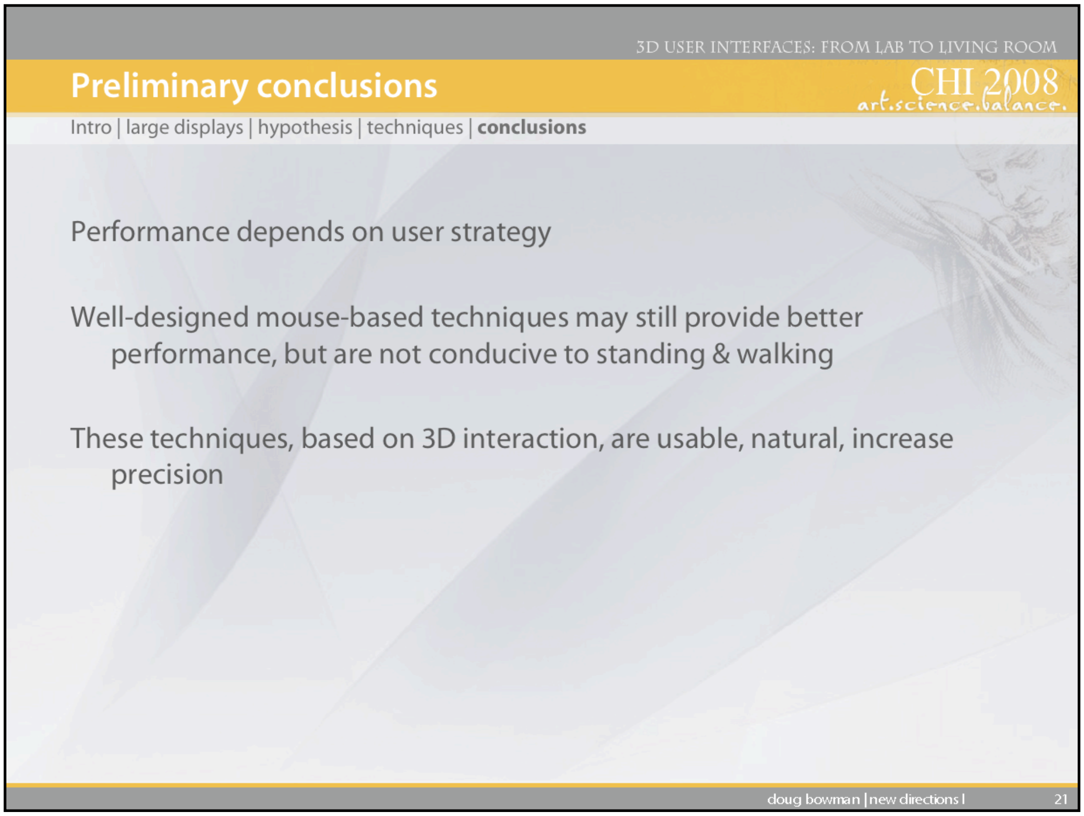
The user can also freeze the interaction context simply by covering one of the tracking markers on the tabletPC with his hand, to allow stable, precise interaction.

Moving an object across the display is easy: the user sets the interaction context, selects the object with the stylus, and then holds the stylus down while pointing the tabletPC at a different area of the large display.

TabletPointing videos

Intro | large displays | hypothesis | **techniques** | conclusions



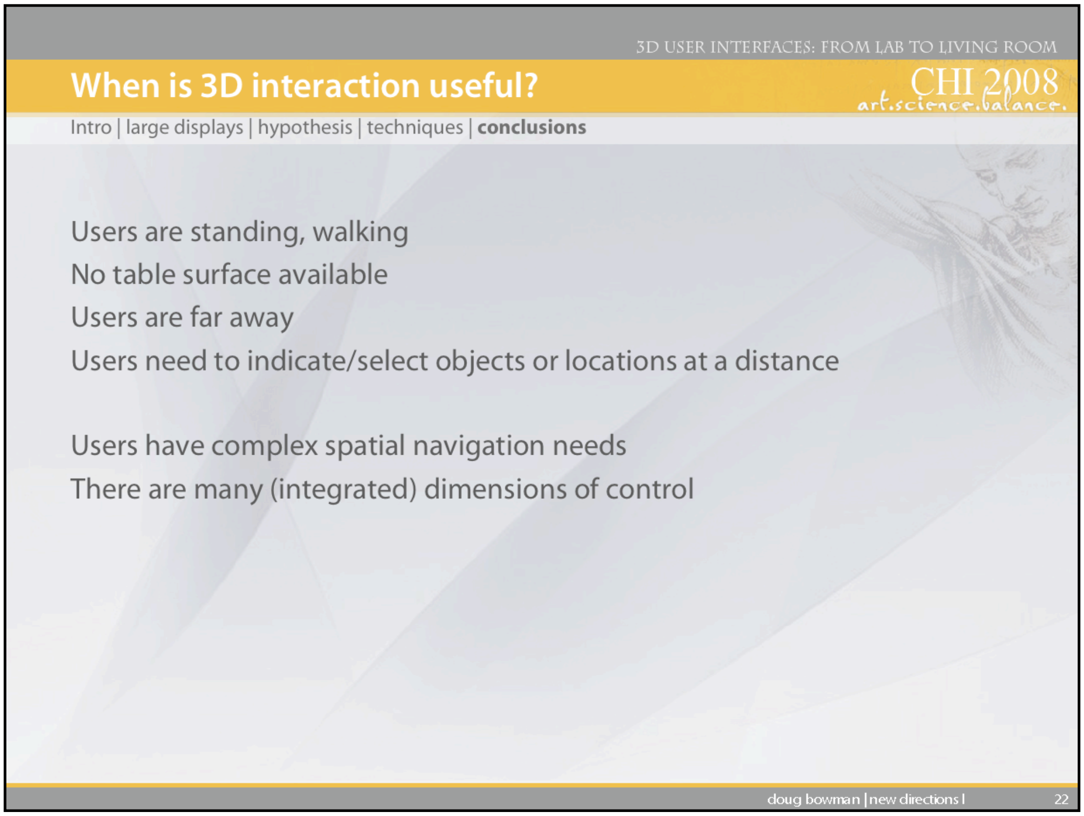


At the time that these notes were printed, we were still in the midst of performing evaluations of these techniques, but here are some preliminary conclusions.

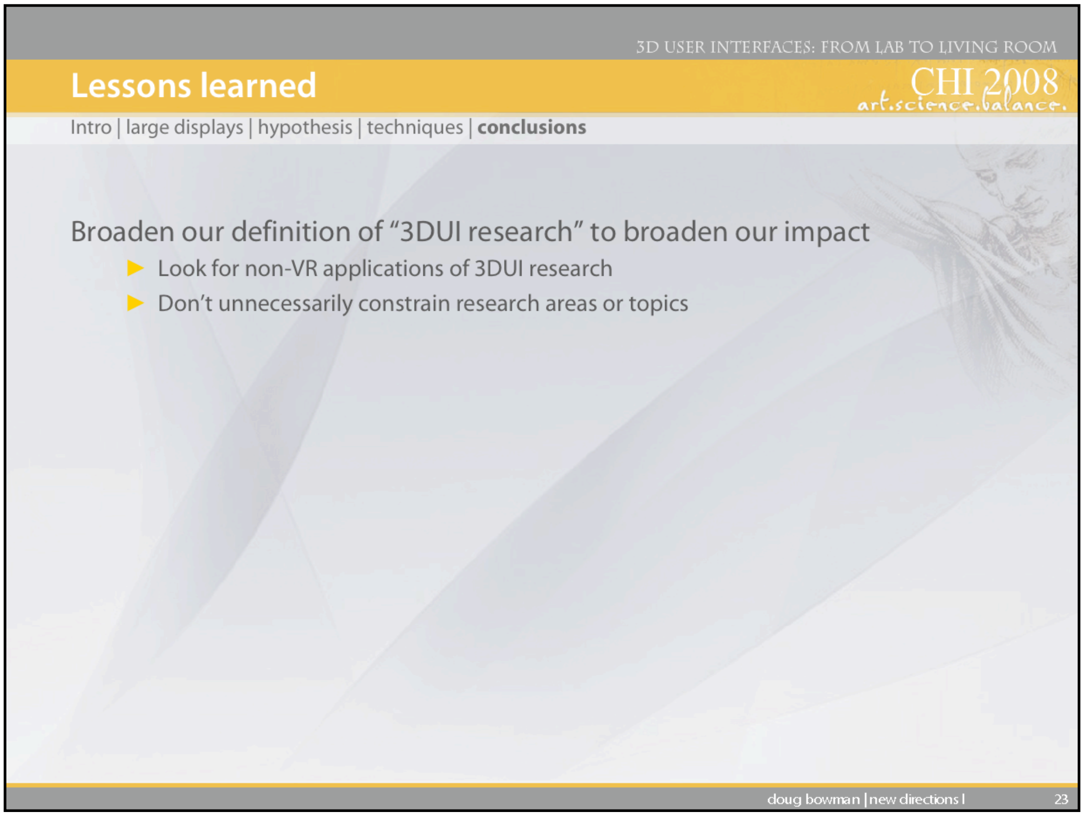
First, user strategy is extremely important for determining the usability and performance of these techniques. Thus, we are evaluating different strategies in our evaluations.

Second, these techniques typically don't yet perform quite as well as well-designed mouse-based techniques. But again, the mouse-based techniques don't lend themselves to standing and walking.

Finally, the techniques do meet their goals of being usable and natural. Cursor tracking is never a problem with these techniques because they use absolute pointing. And when high precision is needed, these techniques can achieve that.



Stepping back from the large display interaction research, we can generalize our experiences to say that 3D interaction may be a reasonable choice when these conditions exist. 3D interaction and 3D UIs are not just for VR and AR.



3D UI researchers should not abandon AR and VR, but we should look for other opportunities to apply our expertise such as the example in this lecture. Other areas like this, such as gaming interfaces, are covered elsewhere in this course.

Acknowledgments

Mara da Silva
Regis Kopper
Mike DellaNoce
Brian Badillo
Ryan McMahan
Will McConnell
Tao Ni
Yi Wang