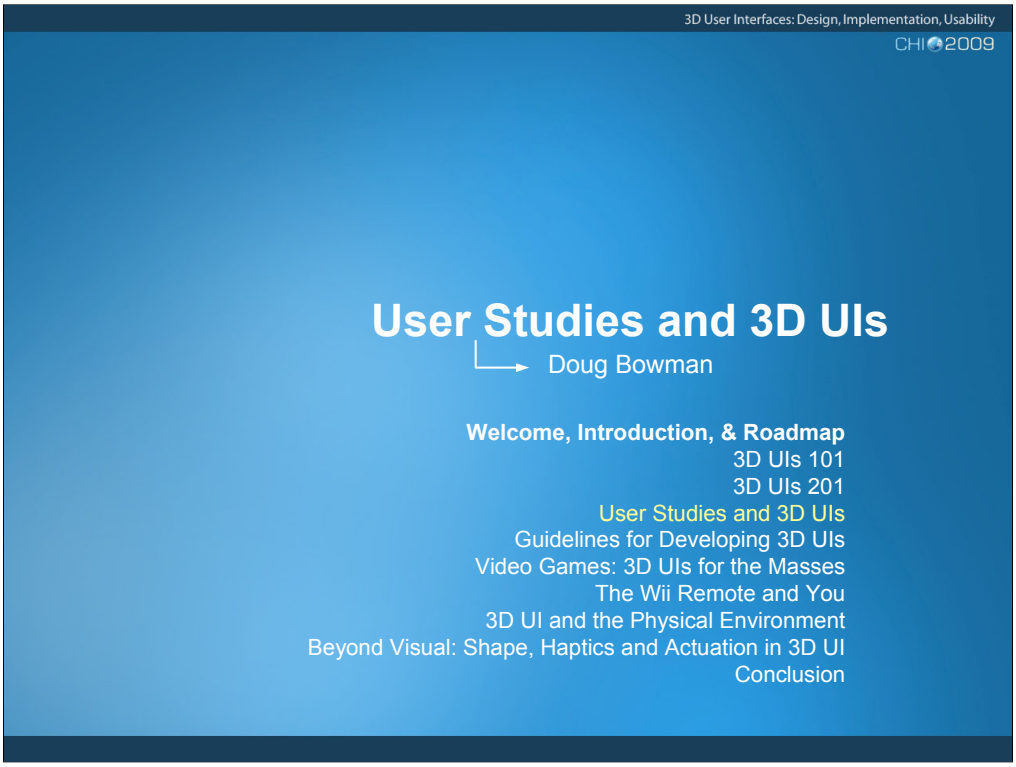


This lecture is on the topic of evaluation of 3D UIs and ways to measure the effectiveness of 3D UIs.

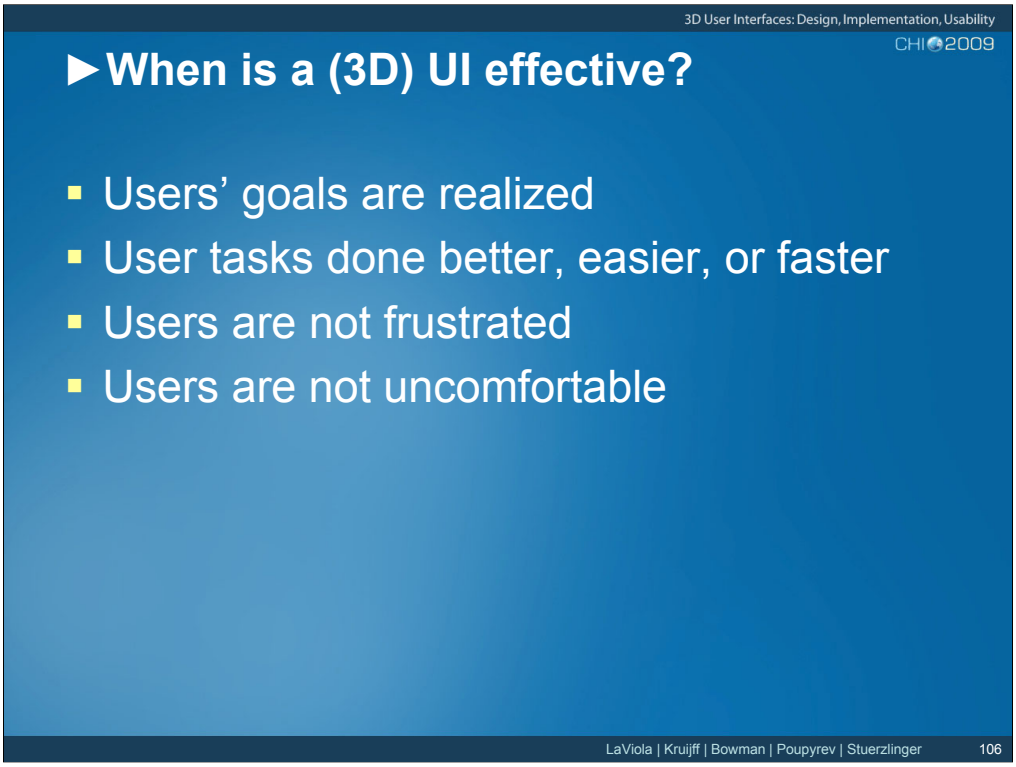
►My background

- Designed, administered, analyzed, or directed many dozens of user studies since 1995
- Major study types:
  - Comparisons of 3D interaction techniques
  - Usability evaluations of VR applications
  - Controlled studies of the effects of various levels of immersion

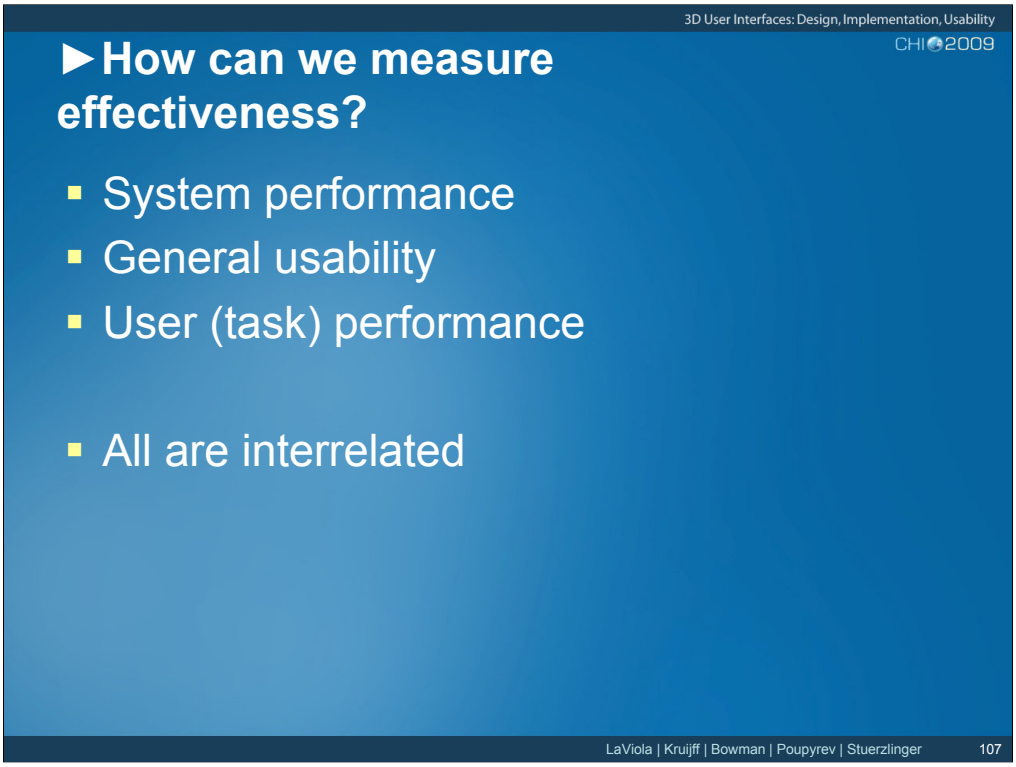
▶Outline

- Metrics for 3D UI evaluation
- Evaluation methods for 3D UIs
- Practical tips for 3D UI user studies

Metrics for 3D UI evaluation



First, we will consider metrics for 3D UIs. That is, how do we measure the characteristics of a 3D UI when evaluating it? I will focus on the general metric of *effectiveness*. A 3D UI is effective when the user can reach her goals, when the important tasks can be done better, easier, or faster than with another system, and when users are not frustrated or uncomfortable. Note that all of these have to do with the *user*. As we will see, typical computer science performance metrics like speed of computation are really not important in and of themselves when we talk about user interfaces. After all, the point of the 3D UI is to serve the needs of the user, so speed of computation is only important insofar as it affects the user's experience or tasks.

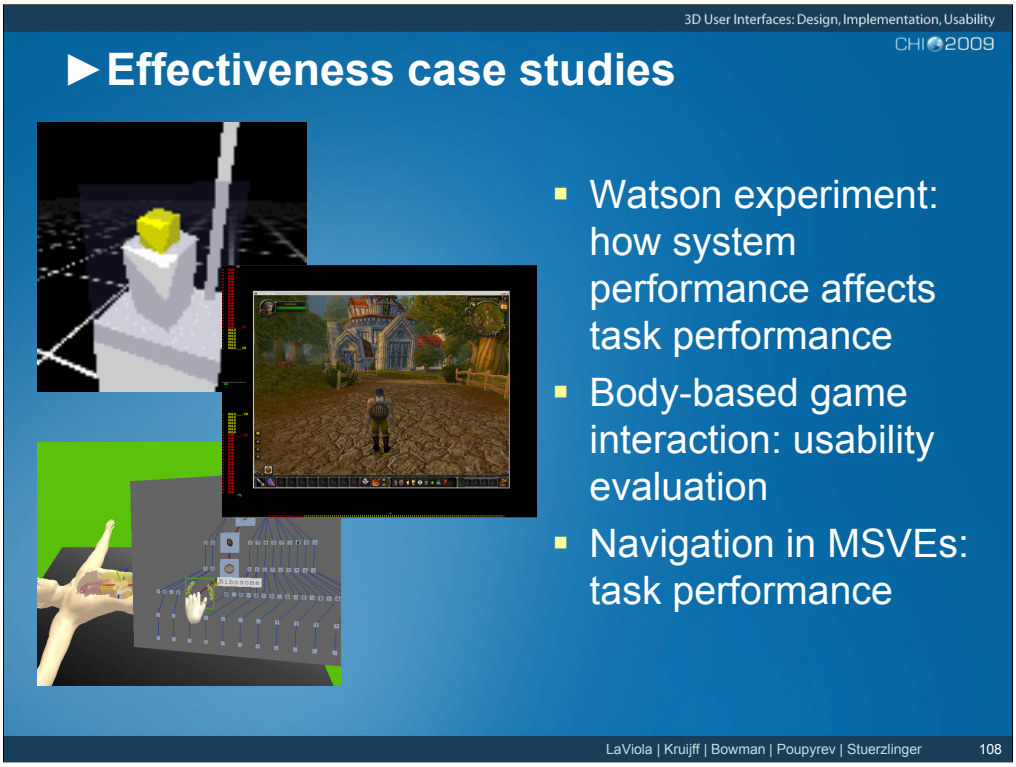


We will talk about three different types of metrics, all of which are interrelated.

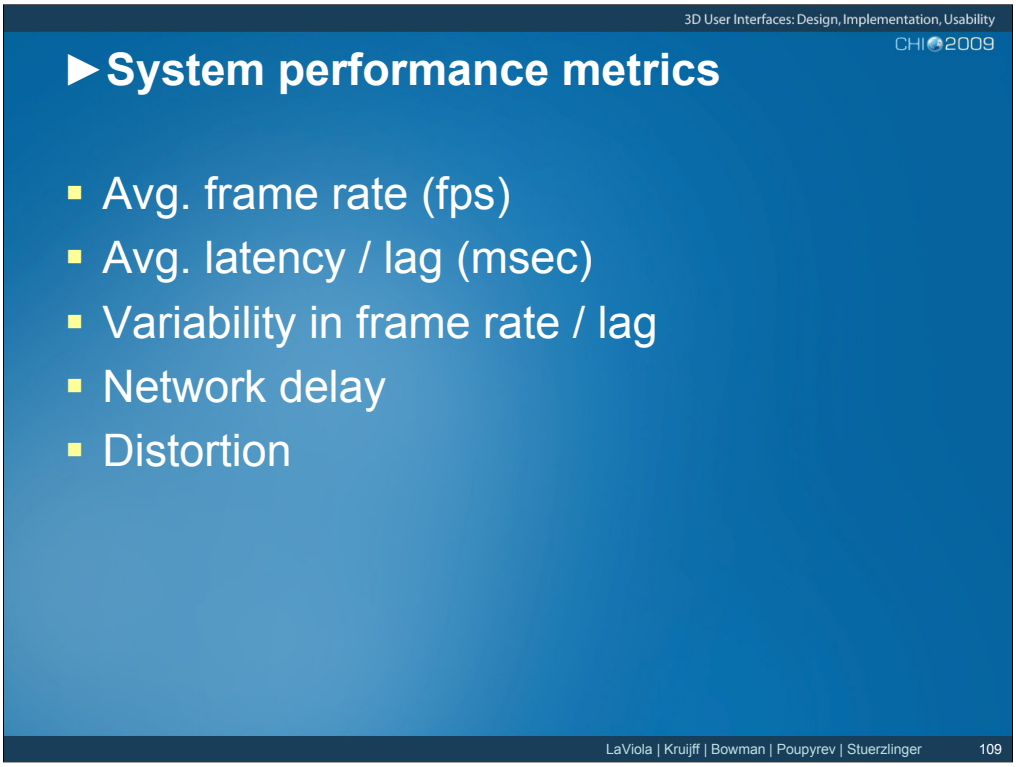
System performance refers to traditional computer science performance metrics, such as frame rate.

General usability refers to traditional HCI metrics like ease of use, ease of learning, satisfaction, critical incidents, etc.

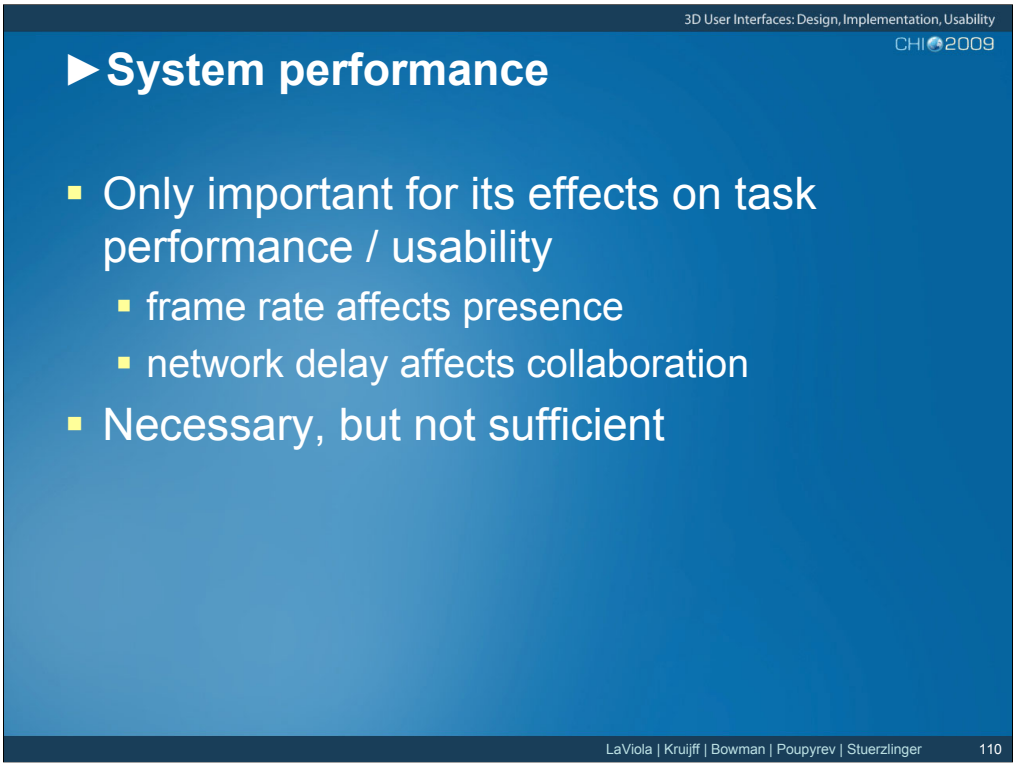
User (task) performance refers to the quality of performance of specific tasks in the 3D UI, such as the time to complete a task.



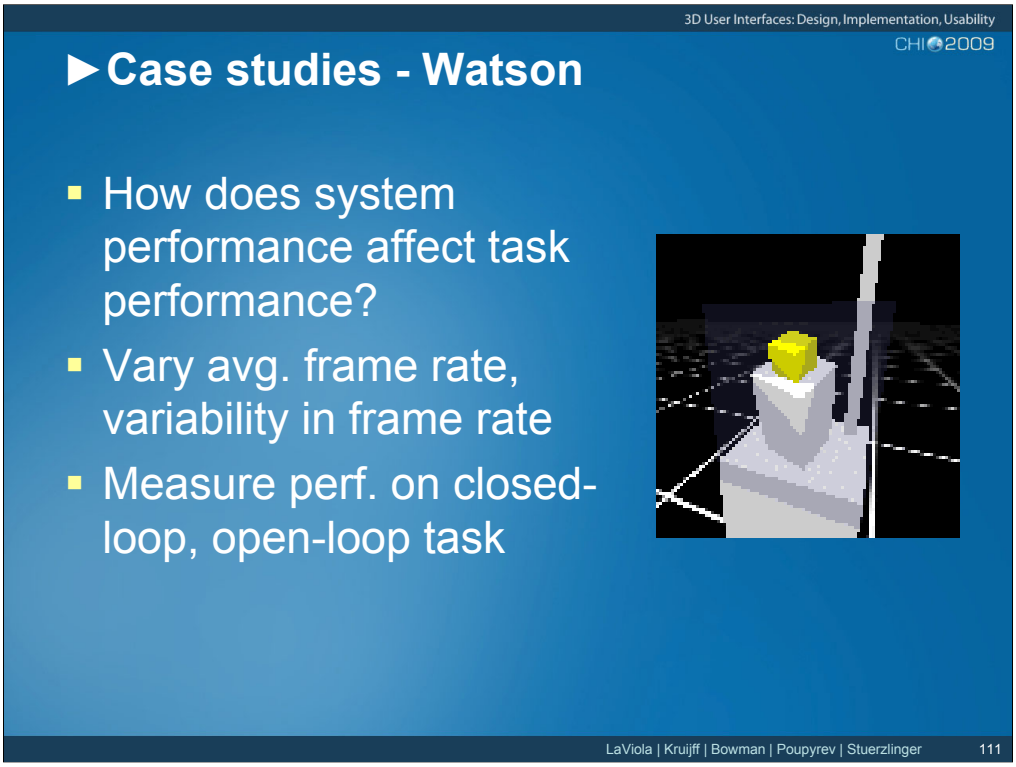
The three types of metrics will be illustrated by three case studies.



Here are some possible system performance metrics for 3D UIs. Note that they are fairly general in nature, that they are measurable, and that most of them apply to any type of 3D UI system or application.



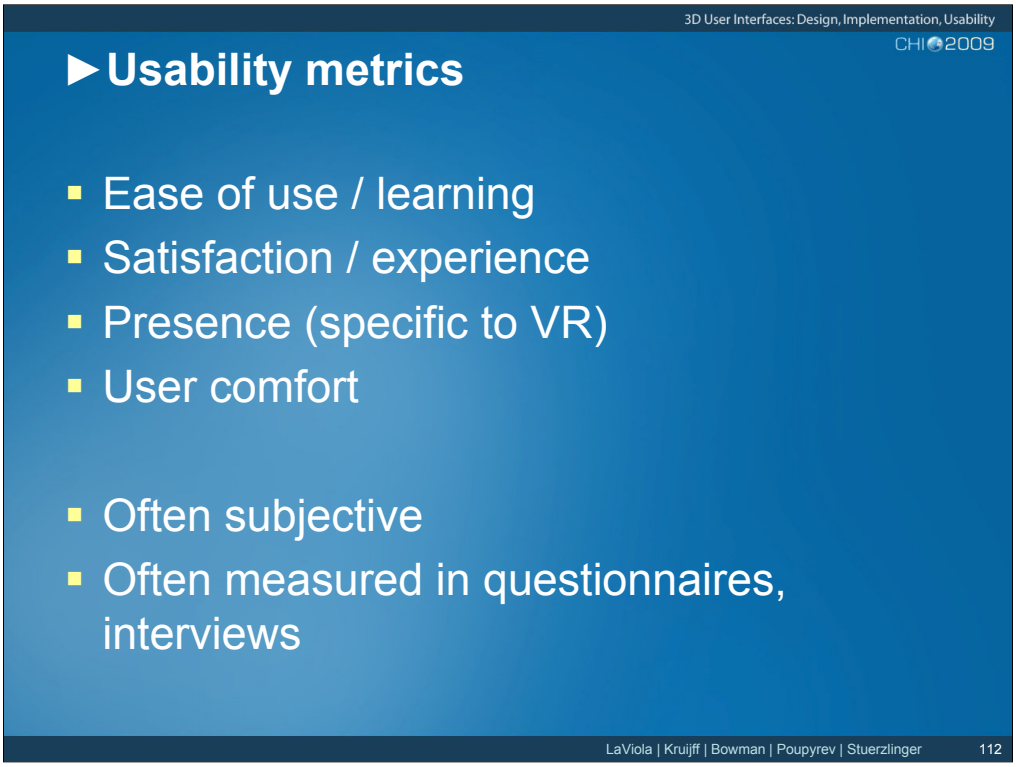
As mentioned earlier, the only reason we're interested in system performance is that it has an effect on interface performance and user performance. For example, the frame rate probably needs to be at "real-time" levels before a user will feel present. Also, in a collaborative setting, task performance will likely be negatively affected if there is too much network delay.



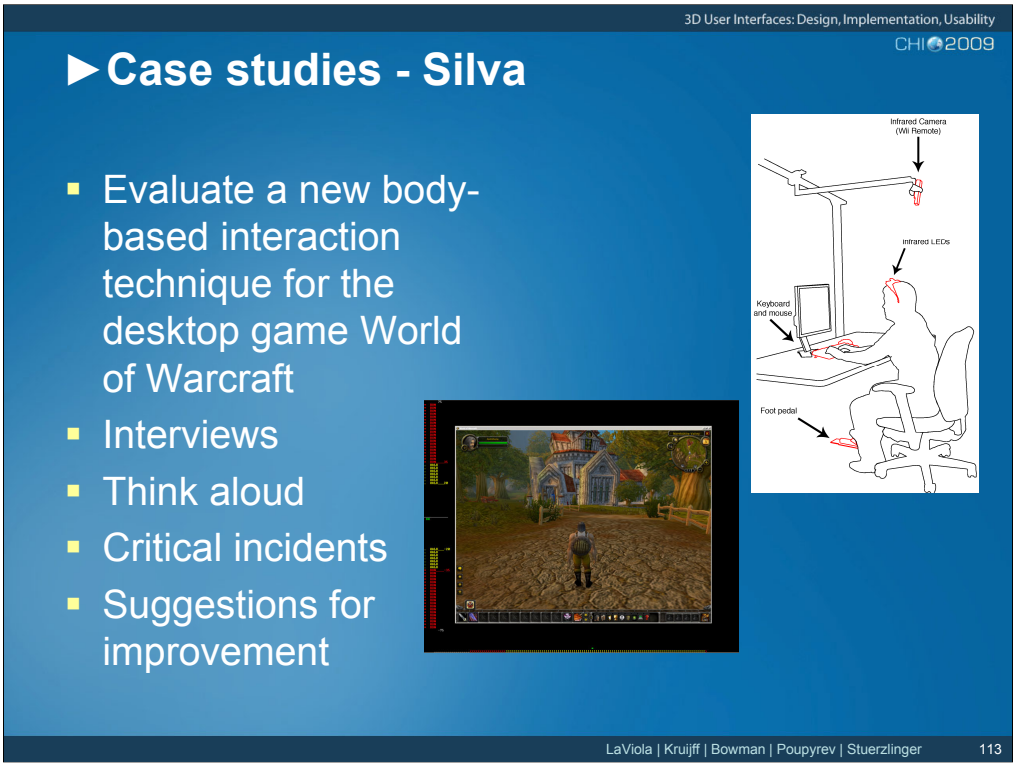
Ben Watson performed some experiments where he varied some system performance values and measured their effect on task performance. For example, one experiment varied the frame rate and also the variability in the frame rate over time. He also looked at different types of tasks. Closed-loop tasks are ones in which users make incremental movements and use visual, proprioceptive, and other types of feedback to adjust their movements during the task. A real-world example is threading a needle. Open-loop tasks do not use feedback during the task – they simply make some initial calculations and then proceed with the movement. A real-world example is catching a ball thrown at a high speed. He found that these tasks were affected in different ways by the various system performance values.

Watson, B., Spaulding, V., Walker, N., and Ribarsky, R. "Evaluation of the Effects of Frame Time Variation on VR Task Performance." Proceedings of the Virtual Reality Annual International Symposium, 38-44, 1997.

B. Watson et al, Effects of variation in system responsiveness on user performance in virtual environments. Human Factors, 40(3), 403-414.

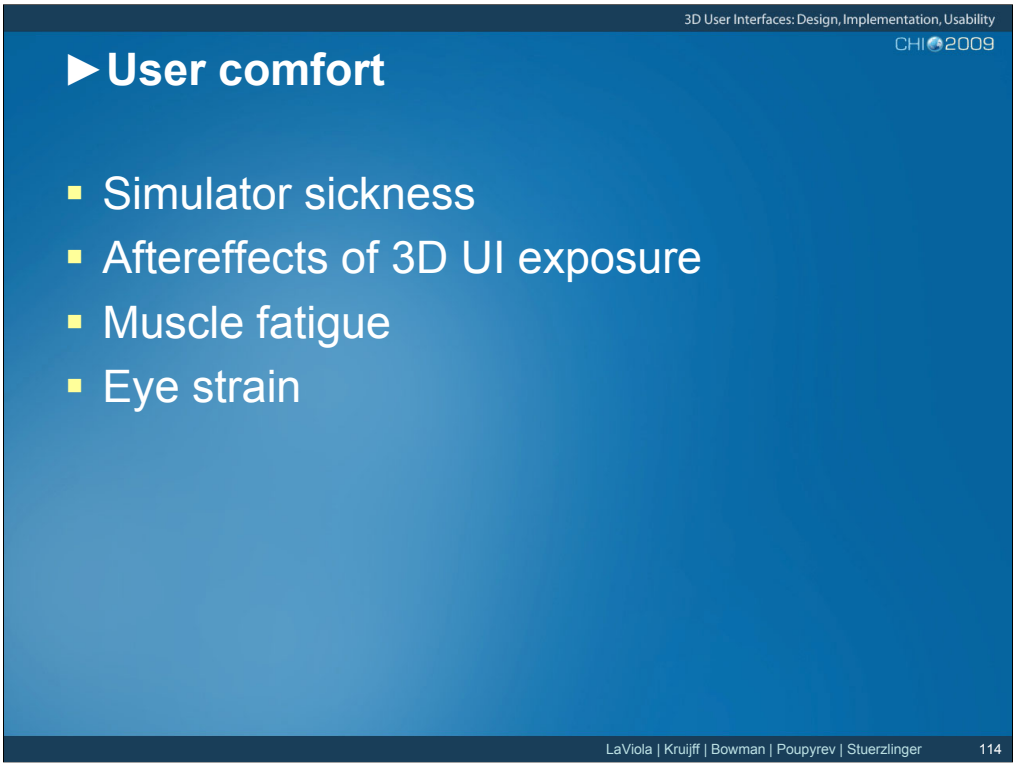


Here are some examples of interface performance (user preference) metrics. These are mostly subjective, and are measured via qualitative instruments, although they can sometimes be quantified. For VR systems in particular, presence and user comfort can be important metrics that are not usually considered in traditional UI evaluation.

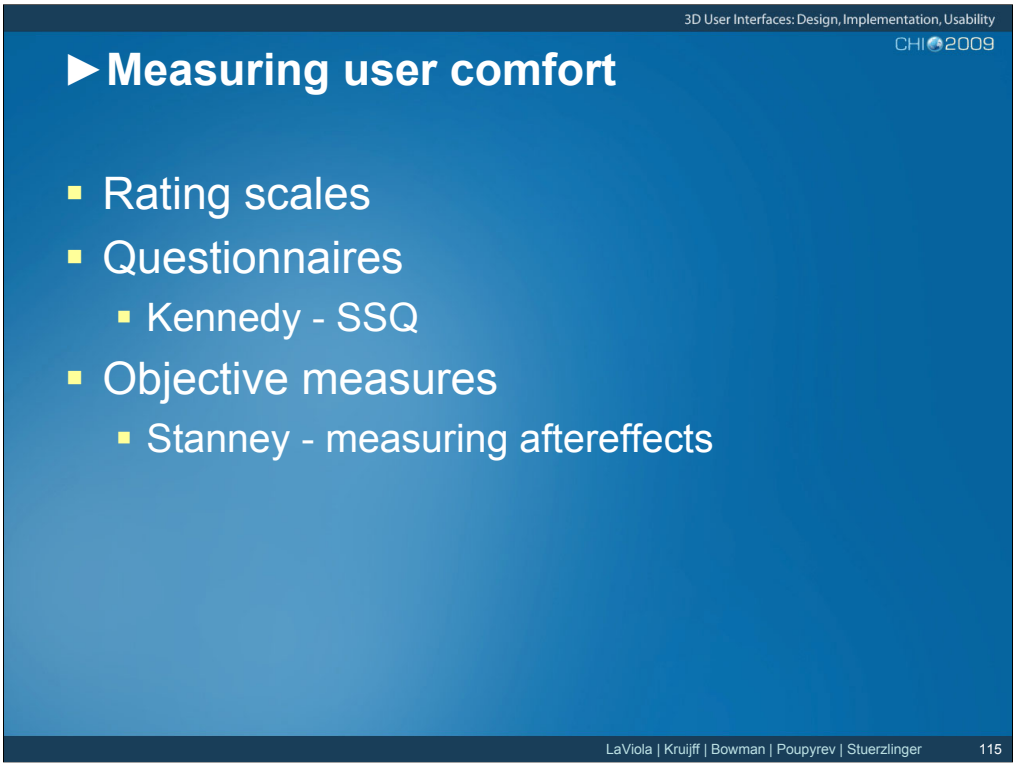


The case study for type of metric involves an evaluation of a novel body-based navigation technique for World of Warcraft. Since the authors wanted to evaluate the feasibility of this approach and the usability of the technique, they gathered users' subjective opinions and observations of user behavior.

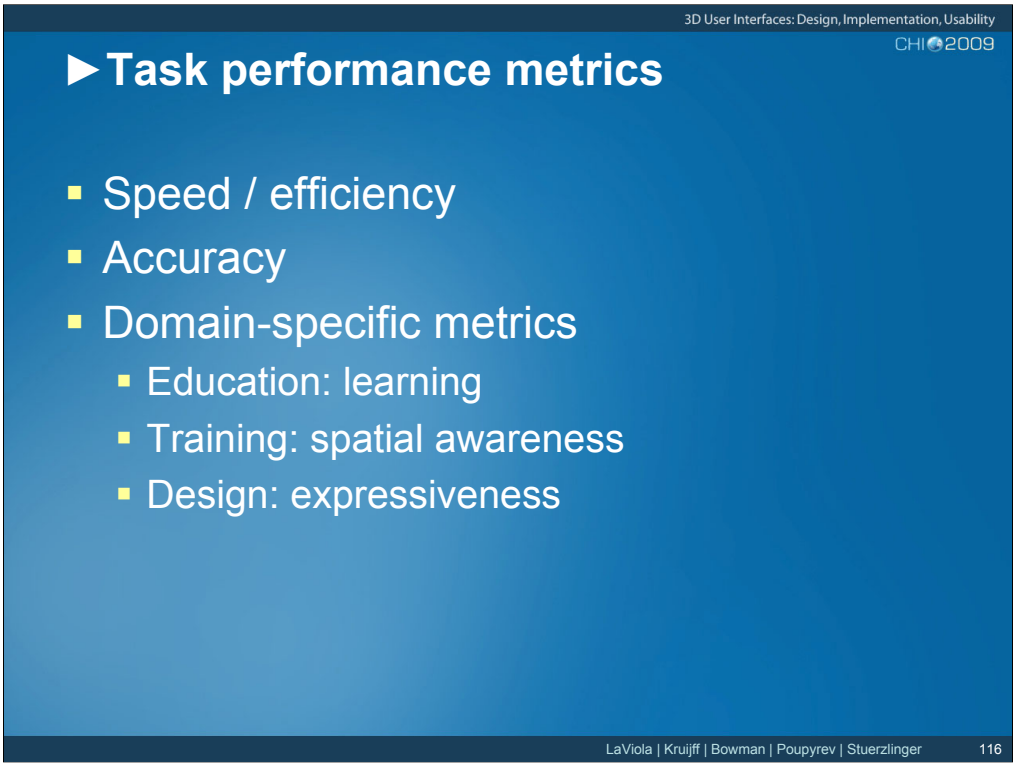
Silva, M. and Bowman, D. Body-based interaction for desktop games. Submitted to CHI 2009 Work-in-Progress.



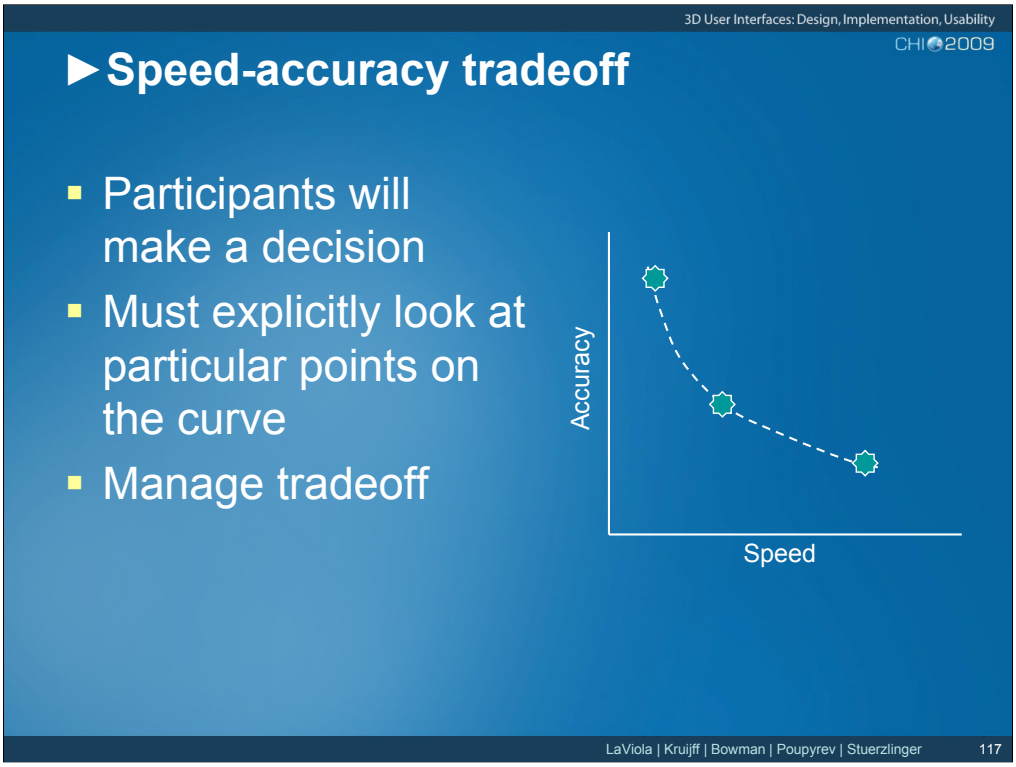
A usability metric that is important for many 3D UIs (but not most traditional interfaces) is user comfort. This includes several different things. The most notable and well-studied is so-called “simulator sickness” (because it was first noted in systems like flight simulators). This is similar to motion sickness, and may result from mismatches in sensory information (e.g. your eyes tell your brain that you are moving, but your vestibular system tells your brain that you are not moving). There is also work on the physical aftereffects of being exposed to 3D UI systems. For example, if a VE mis-registers the virtual hand and the real hand (they’re not at the same physical location), the user may have trouble doing precise manipulation in the real world after exposure to the virtual world. More seriously, things like driving or walking may be impaired after extremely long exposures (1 hour or more). Finally, there are issues with muscle fatigue (e.g., a UI with a handheld 3D input device may require constant muscle tension) or eye strain (e.g., because of incorrect oculomotor cues in 3D stereoscopic displays) from the use of 3D UI hardware.



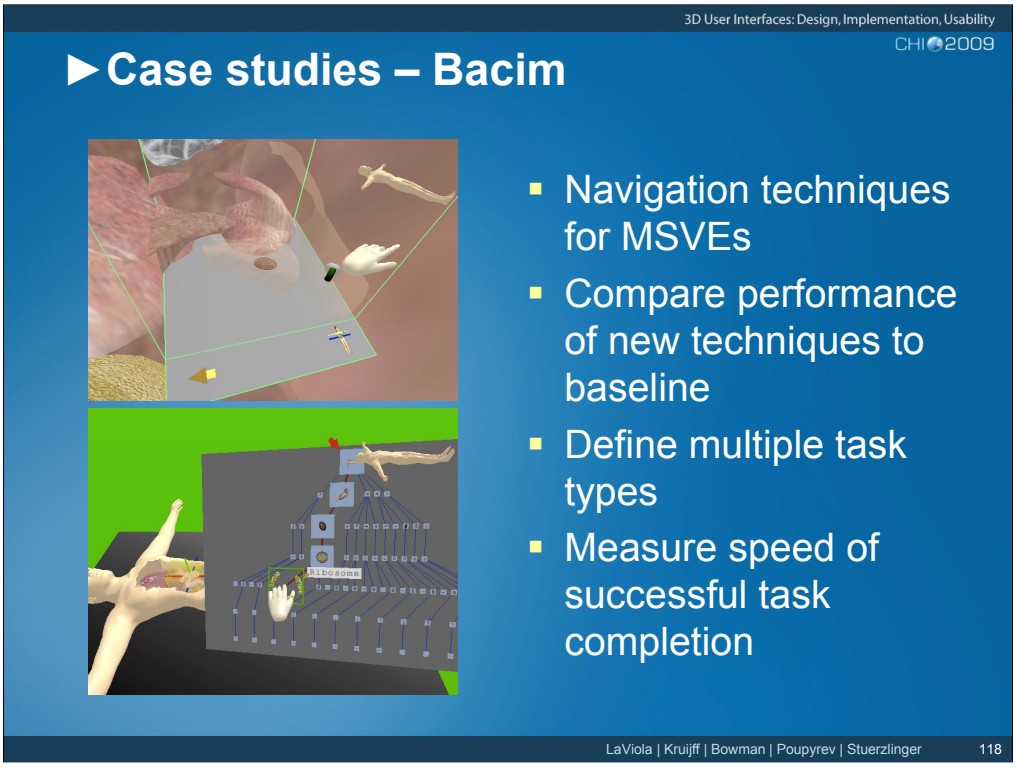
User comfort is also usually measured subjectively, using rating scales or questionnaires. The most famous questionnaire is the simulator sickness questionnaire (SSQ) developed by Robert Kennedy. Kay Stanney has attempted some objective measures in her study of aftereffects – for example by measuring the accuracy of a manipulation task in the real world after exposure to a virtual world.



The last category of metrics are task performance metrics. These include general measures like speed and accuracy, and domain-specific measures such as learning and spatial awareness.



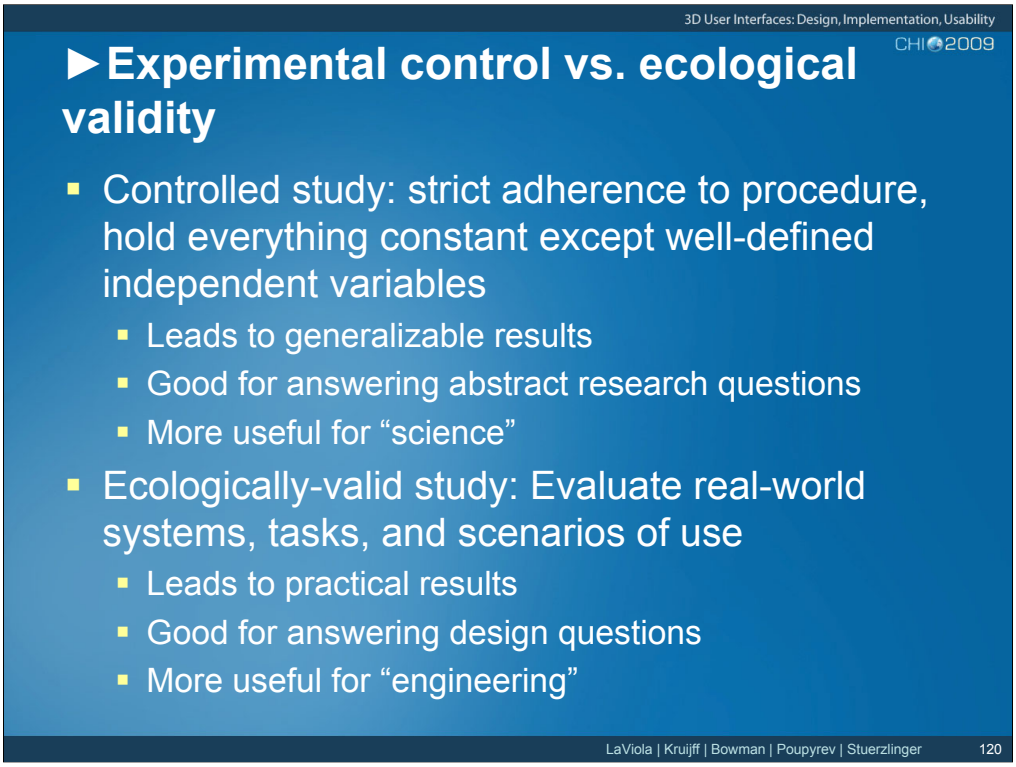
The problem with measuring speed and accuracy is that there is an implicit relationship between them: I can go faster but be less accurate, or I can increase my accuracy by decreasing my speed. It is assumed that for every task there is some curve representing this speed/accuracy tradeoff, and users must decide where on the curve they want to be (even if they don't do this consciously). So, if I simply tell my participants to do a task as quickly and precisely as possible, they will probably end up all over the curve, giving me data with a high level of variability. Therefore, it is very important that you instruct users in a very specific way if you want them to be at one end of the curve or the other. Another way to manage the tradeoff is to tell users to do the task as quickly as possible one time, as accurately as possible the second time, and to balance speed and accuracy the third time. This gives you information about the tradeoff curve for the particular task you're looking at.



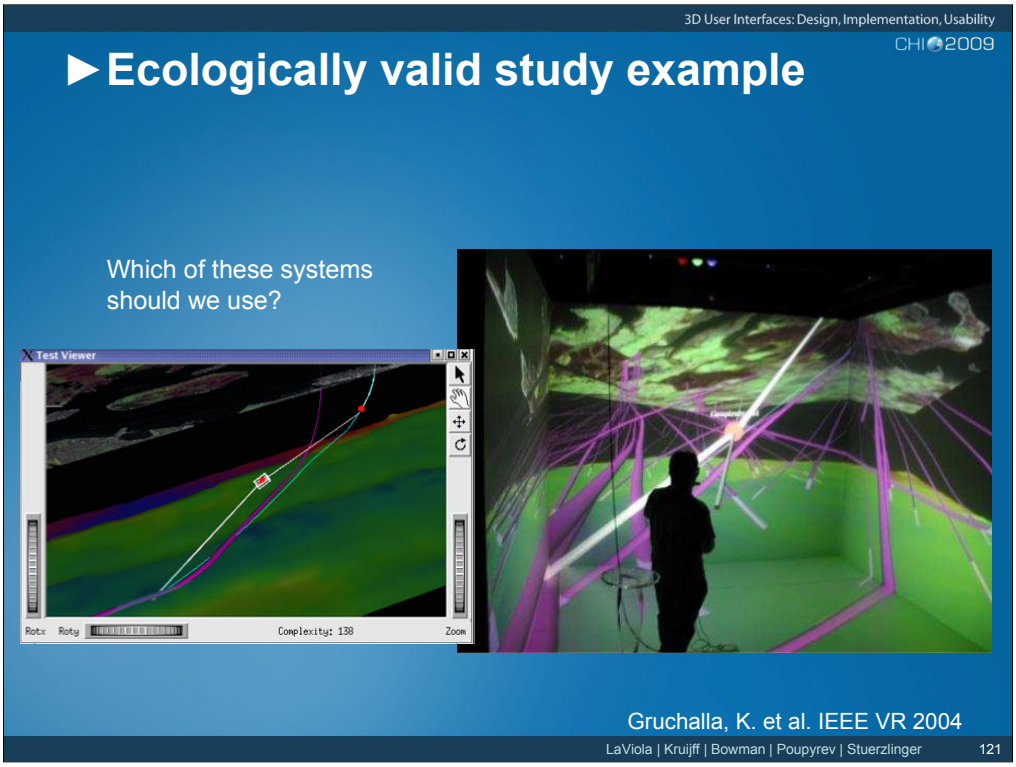
The case study for task performance metrics is a recent project involving navigation techniques for multi-scale virtual environments (MSVEs). The authors designed two new techniques meant to improve the amount and quality of wayfinding information that is provided to users of these complex environments. To evaluate the effectiveness of these techniques (after verifying their usability), they compared user task performance on multiple basic task types using the new techniques and a pre-existing set of baseline techniques. To manage the speed-accuracy tradeoff, they required that all tasks be completed successfully, and measured speed.

Bacim, F., Bowman, D., and Pinho, M. Wayfinding Techniques for MultiScale Virtual Environments. Proceedings of the IEEE Symposium on 3D User Interfaces, 2009.

Evaluation methods for 3D UIs

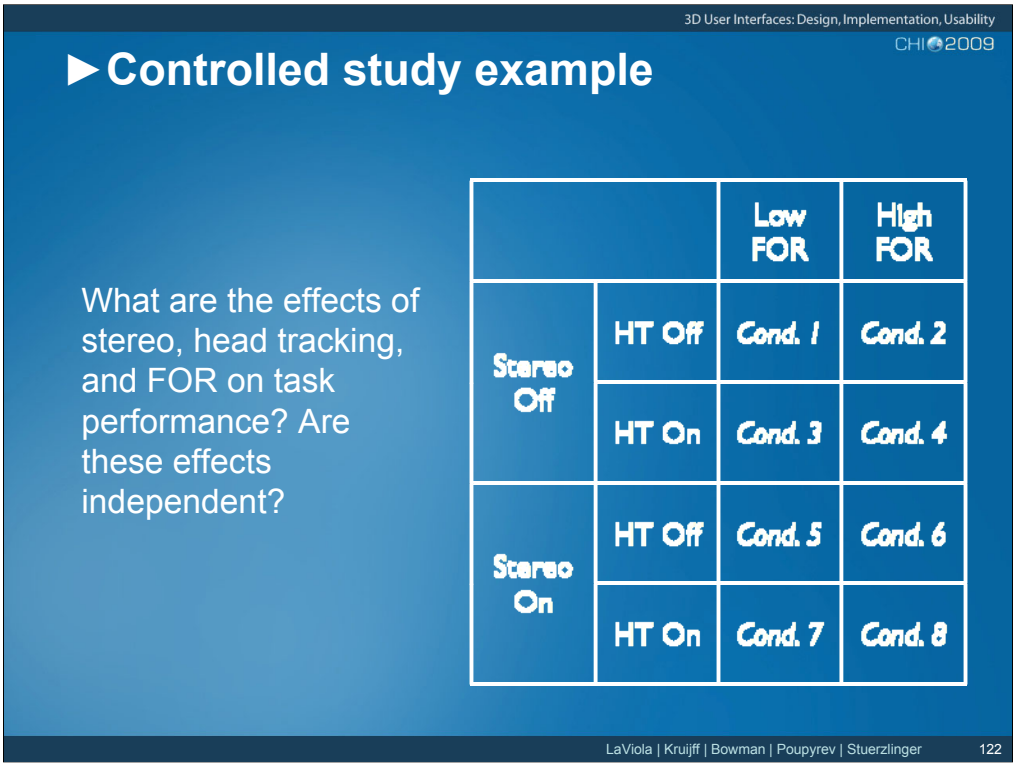


As in many experimental settings, there is often a tension and tradeoff in 3D UI evaluation between experimental control and ecological validity. The more controlled a study is, the less similar to the real world use of the 3D UI it becomes. If real-world usage is evaluated, it leads to a lack of experimental control. There is actually a continuum between these two extremes. Where a study falls on the continuum depends on its goals and on the current state of knowledge about the topic of the study.



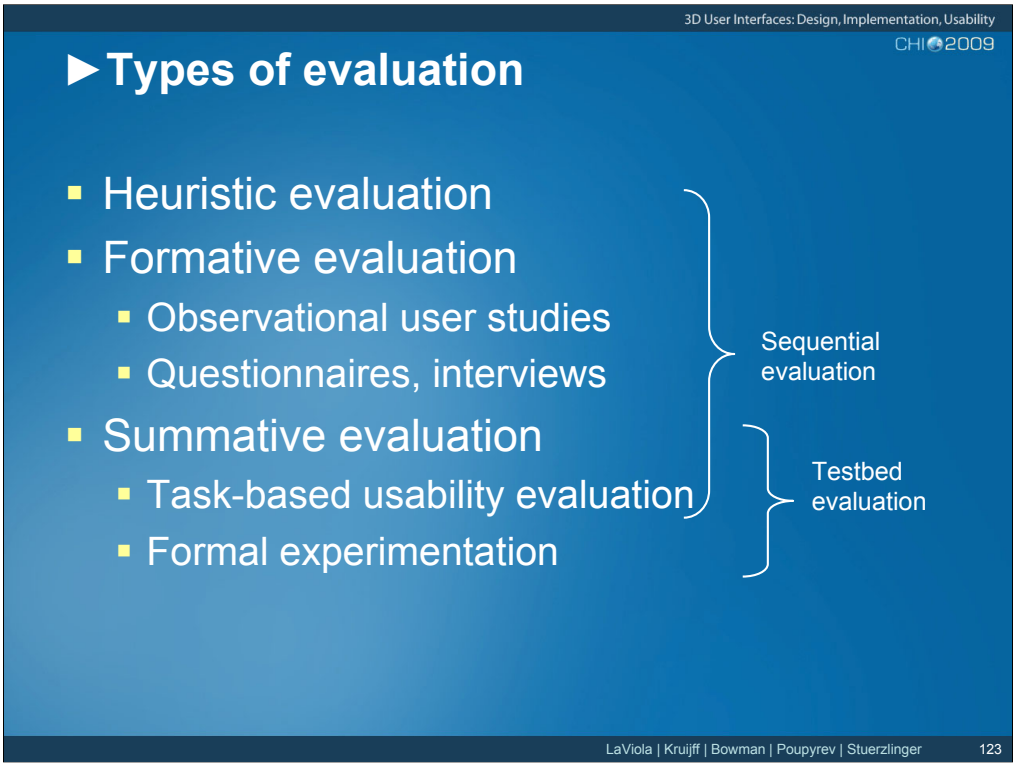
In this ecologically-valid study, the authors asked users to perform tasks using two versions of a real-world application: one that ran on the desktop and another that ran in a CAVE. The results clearly showed which application was best for each task, but did not generalize.

Gruchalla, K. "Immersive Well-Path Editing: Investigating the Added Value of Immersion." IEEE Virtual Reality, Chicago, 157-164, 2004.



On the other hand, we could run a more controlled study on similar systems by varying independent factors that describe the 3D UIs used in the real-world applications. For example, we could vary field of regard (FOR), stereo, and head tracking (HT) independently with two levels each, using the same physical displays and input devices for each condition. A study of this sort can be seen in:

Raja, D. (2006). "The Effects of Immersion on 3D Information Visualization," Masters Thesis, Dept. of Computer Science, Virginia Tech, Blacksburg, Virginia.



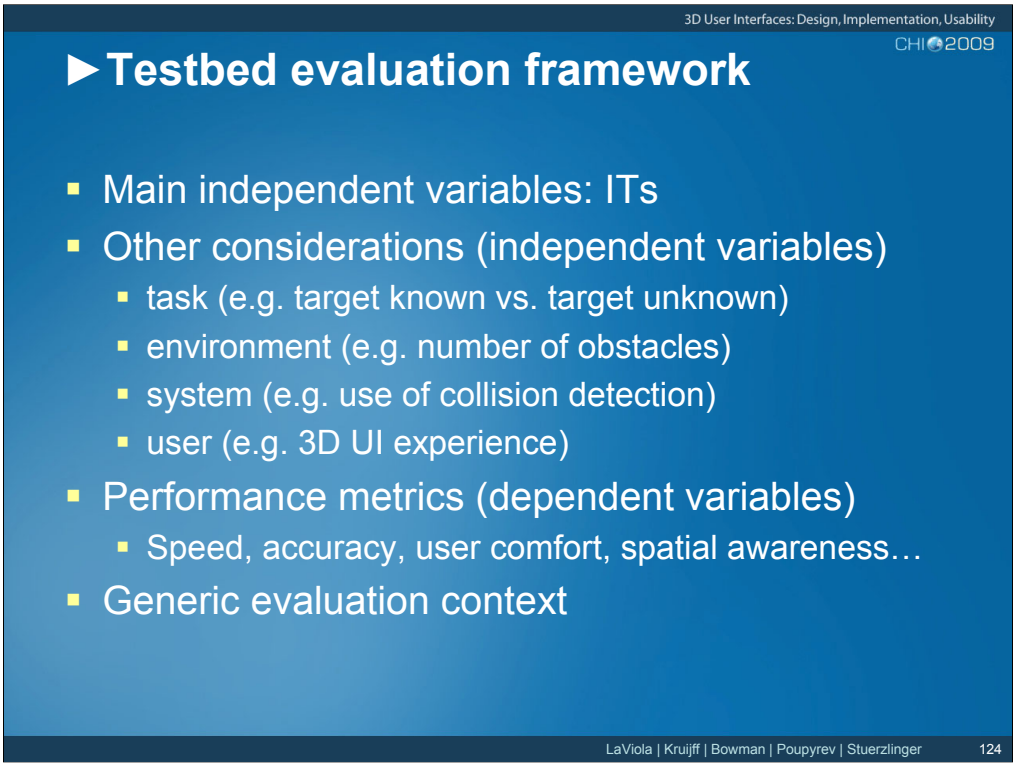
Here are some general categories of user interface evaluation methods that are applicable to 3D UIs.

Heuristic evaluation refers to an evaluation by interface experts, using a well-defined set of heuristics or guidelines. Experts examine the interface visually, via a written description, or through actual use, and determine whether or not the interface meets the criteria set forth in the heuristics. For example, the interface might be checked to see if it meets the guideline: “Eliminate extraneous degrees of freedom for a manipulation task.”

Formative evaluations are used to refine the design of a widget, an interaction technique, or a UI metaphor. Observational user studies are informal sessions in which users try out the proposed interface. They may be asked to simply explore and play around, or to do some simple tasks. Often users’ comments are recorded (“think out loud” or verbal protocol), and the evaluator watches the user to see if there are parts of the interface that are frustrating or difficult. Post-hoc questionnaires and interviews may be used to get more detailed information from users about their experiences with the system.

Summative evaluations compare various techniques in a single experiment. A task-based usability evaluation is more structured. Users are given specific tasks to perform. Often, users are timed as they perform the tasks, and evaluators may keep track of errors made by the user. This information is then used to improve the interface. Formal experiments have a formal design including independent and dependent variables, subjects from a particular subject pool, a strict experimental procedure, etc. The results of formal experiments are usually quantitative, and are analyzed statistically.

We will be talking about two specific evaluation approaches in this section. Sequential evaluation spans a wide range of evaluation types. Testbed evaluation involves summative techniques.



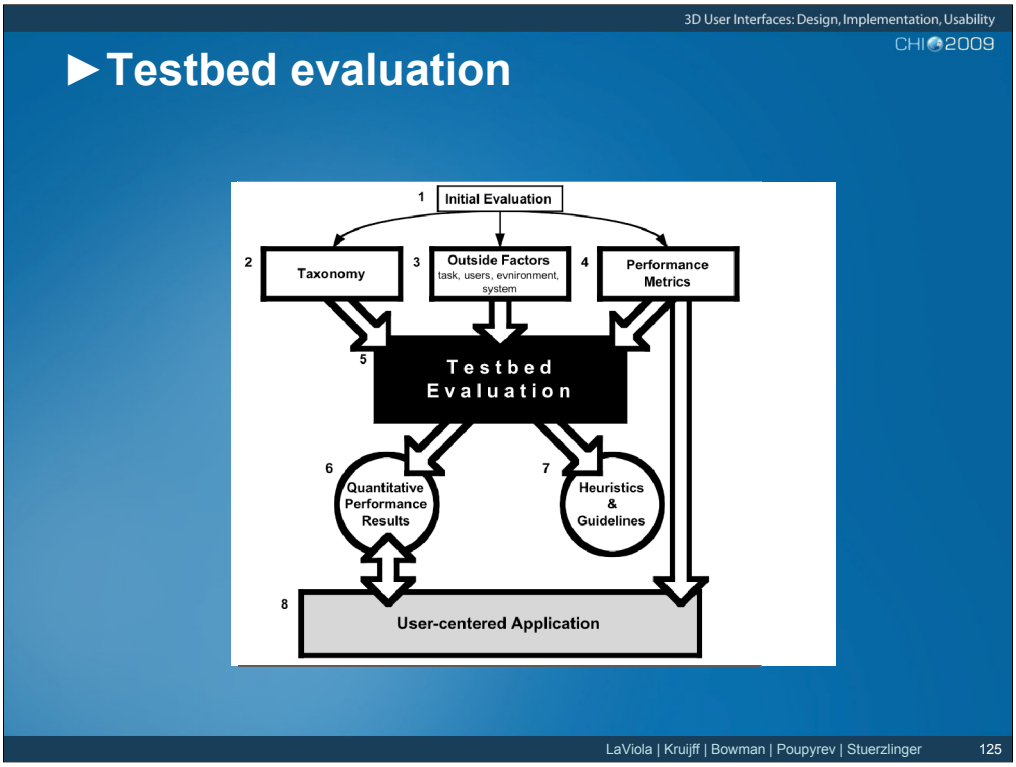
An evaluation testbed is a generalized environment in which many smaller experiments or one large experiment can be run, covering as much of the design space as you can. Like other formal experiments, you're evaluating interaction techniques (or components), but you also include other independent variables that could have an effect on performance. These include characteristics of the task, environment, system, and user.

You also measure multiple dependent variables in such experiments to try to get a wide range of performance data. Here we use performance in the broader sense, not just meaning quantitative metrics. The more metrics you use, the more applications can use the results of the experiment by listing their requirements in terms of the metrics, then searching the results for technique(s) that meet those requirements.

Doug Bowman performed such evaluations in his doctoral dissertation, available online at: http://www.cs.vt.edu/~bowman/thesis/. A summary version of these experiments is in this paper:

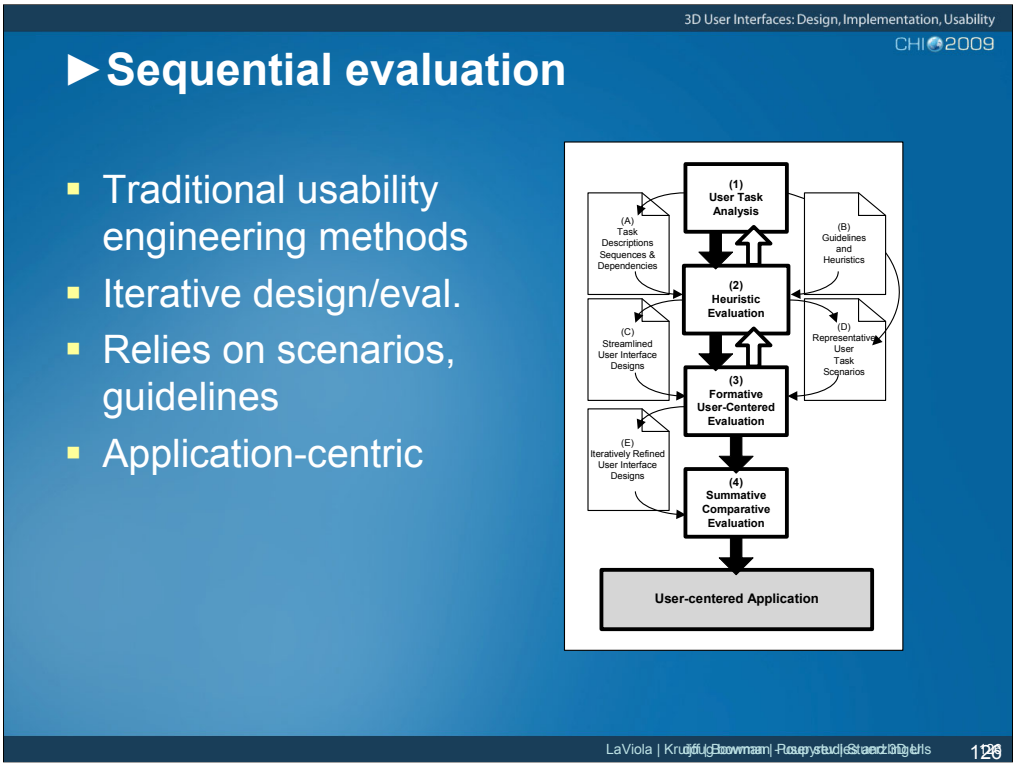
Bowman, Johnson, & Hodges, Testbed Evaluation of VE Interaction Techniques, Proceedings of ACM VRST '99

Also see: Poupyrev, Weghorst, Billinghurst, and Ichikawa, A Framework and Testbed for Studying Manipulation Techniques, Proceedings of ACM VRST '97.



This figure shows the process used in testbed evaluation. Before designing a testbed, one must understand thoroughly the task(s) involved and the space of interaction techniques for those tasks. This understanding can come from experience, but it's more likely to come from some initial (usually informal) evaluations. This can lead to a taxonomy for a task, a set of other factors that are hypothesized to affect performance on that task, and a set of metrics (discussed later).

These things are then used to design and implement a testbed experiment or set of experiments. The results of running the testbed are the actual quantitative results, plus a set of guidelines for the usage of the tested techniques. The results can be used many times to design usable applications, based on the performance requirements of the application specified in terms of the performance metrics.



A different approach is called sequential evaluation, which is a 3D UI-specific form of usability engineering. See the paper:

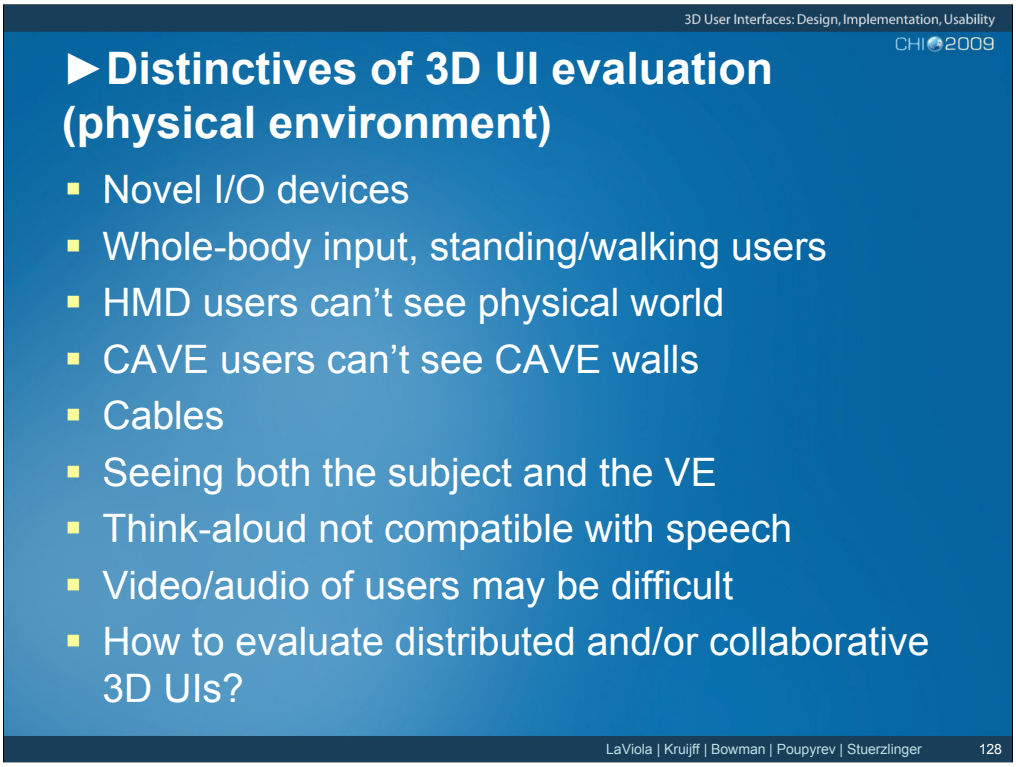
Gabbard, J. L., Hix, D, and Swan, E. J. (1999). User Centered Design and Evaluation of Virtual Environments , *IEEE Computer Graphics and Applications*, *19*(6), 51-59.

As the name implies, this is actually a set of evaluation techniques run in sequence. The techniques include user task analysis, heuristic evaluation, formative evaluation, and summative evaluation. As the figure shows, the first three steps can also involve iteration. Note that just as in testbed evaluation, the goal is a user-centered application.

Sequential evaluation uses both experts and users, produces both qualitative and quantitative results, and is application-centric.

Note that neither of these evaluation approaches is limited to being used for the evaluation of 3D UIs. However, they do recognize that applications with 3D UIs require a more rigorous evaluation process than traditional 2D UIs, which can often be based solely on UI guidelines.

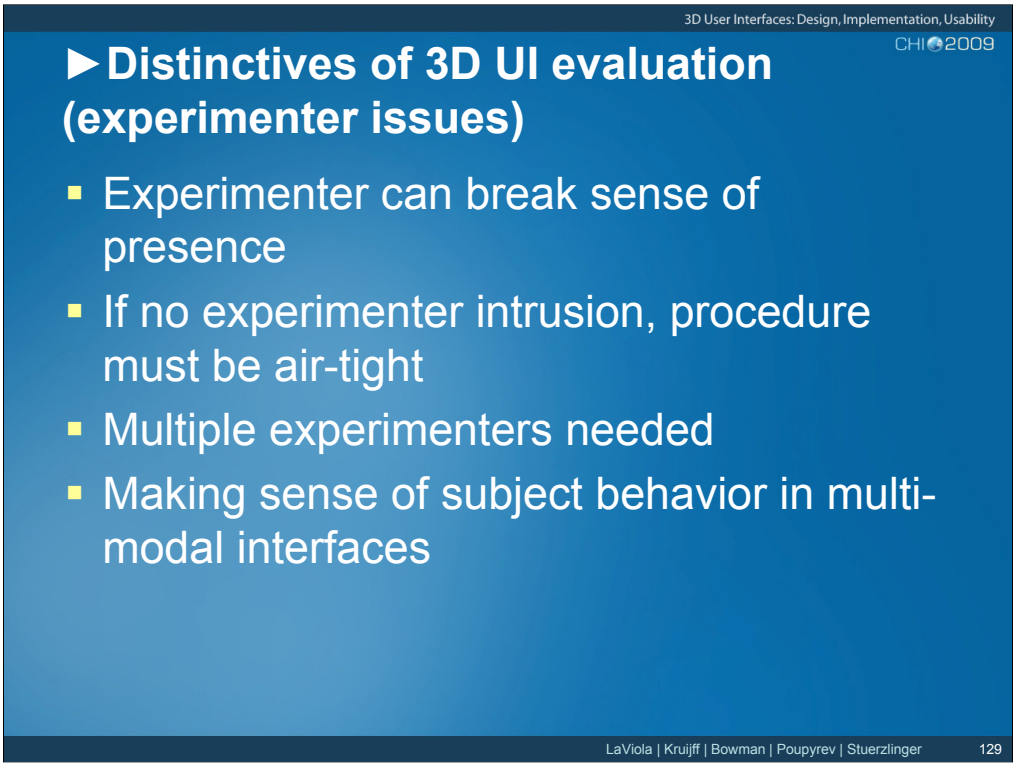
Practical tips for 3D UI user studies



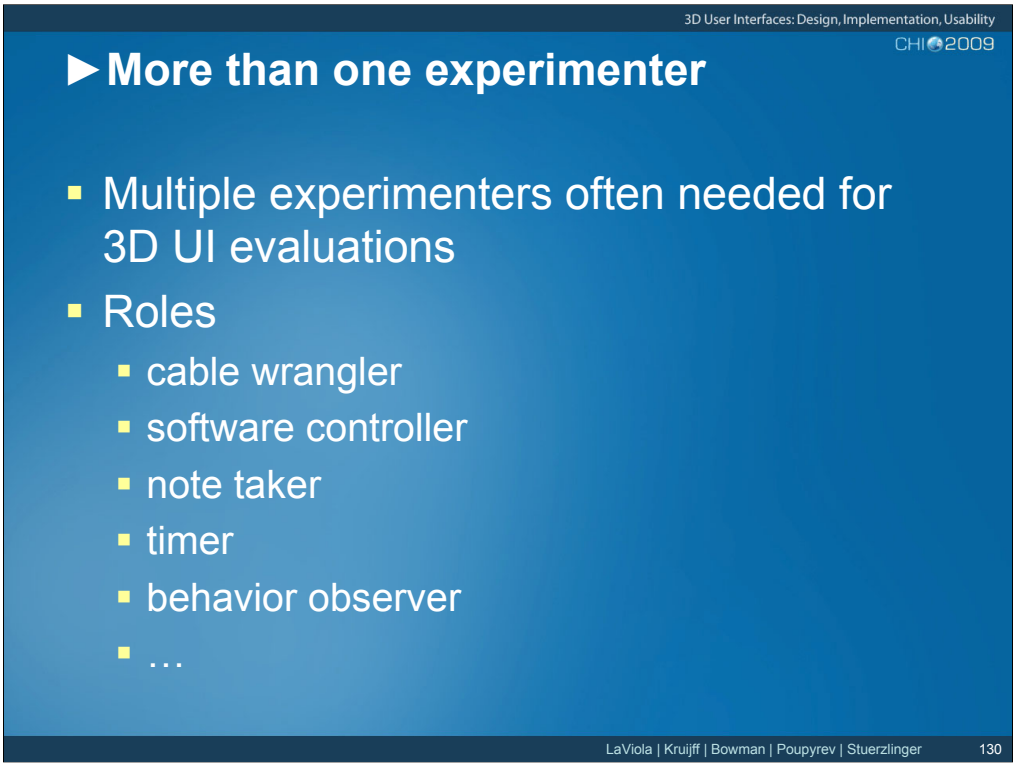
The first practical tip is to acknowledge and plan for the differences between traditional evaluations and 3D UI evaluations. There are a number of ways in which evaluation of 3D interfaces is different from traditional user interface evaluation.

First, there are physical issues. For example, in an HMD-based VE, the physical world is blocked from the user's view. This means that the evaluator must ensure that the user does not bump into objects or walls, that the cables stay untangled, and so on. Another example involves a common method in traditional evaluation called a "think-aloud protocol". This refers to a situation in which the user talks aloud about what he is thinking/doing in the interface. However, many 3D applications use speech input, which of course is incompatible with this evaluation method unless there is an explicit "push-to-talk" technique. Even in this case, the user could not invoke a command while simultaneously describing his thoughts/actions to the evaluator.

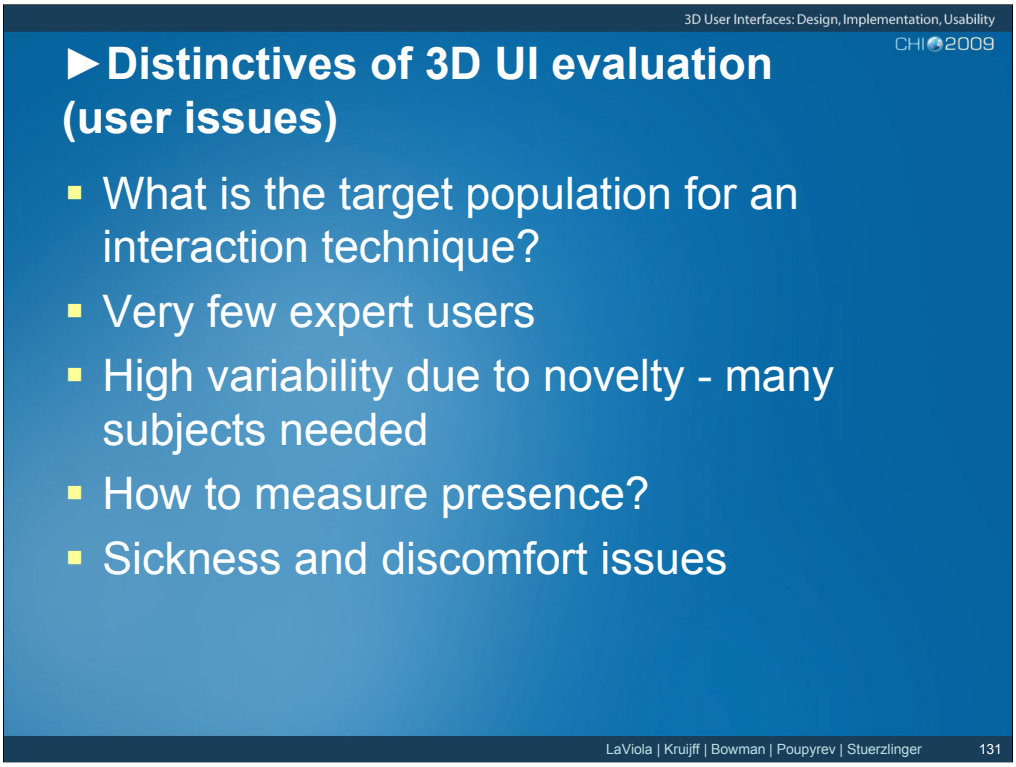
Many other issues of this type are described in our book, 3D User Interfaces: Theory and Practice



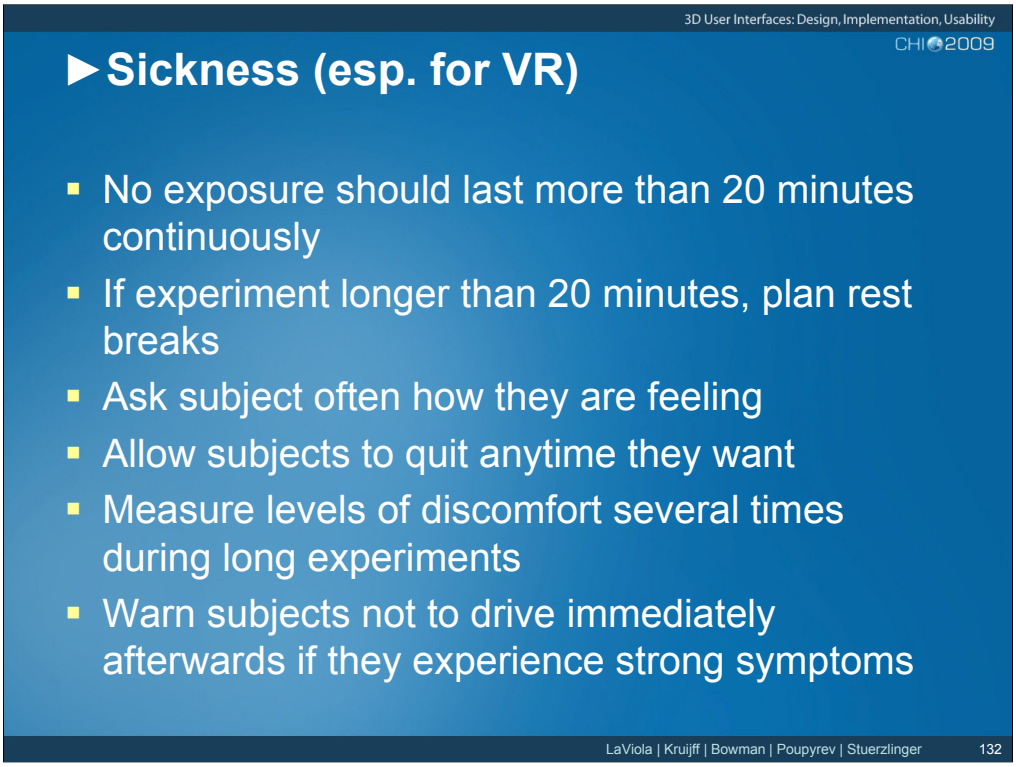
Second, we consider issues related to the experimenter. One of the most important is that an evaluator can break the user's sense of presence by talking to the user, touching the user, making changes to the environment, etc. during an evaluation. If the sense of presence is considered important to the task/application, the evaluator should try to avoid contact with the user during the tests. Another example of an evaluator issue is that multiple evaluators are usually needed. This is because 3D systems are so complex (hardware and software) and because users of 3D UIs have much more freedom and input bandwidth than users of a traditional UI.



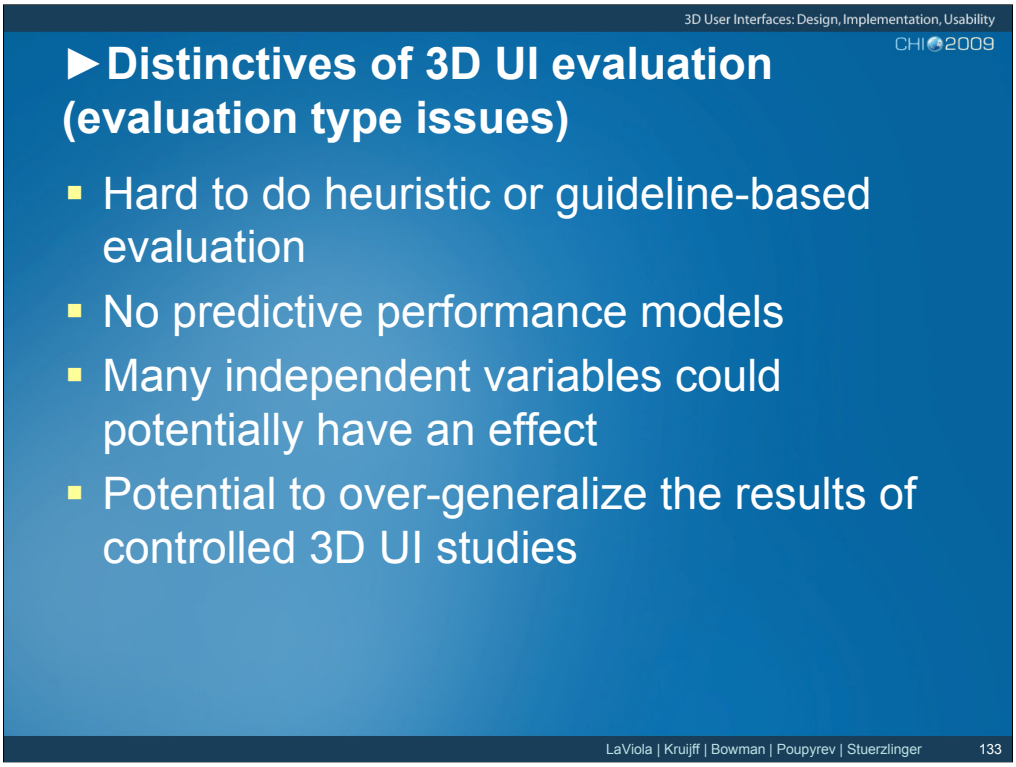
Highlighting one of these issues from the previous slide … Many 3D UI evaluations are more complicated to run than traditional interface studies, because of the complexity of displays, input devices, and software, and because of the number of types of data you may want to measure simultaneously. Thus, it's useful to have multiple experimenters rather than just one.



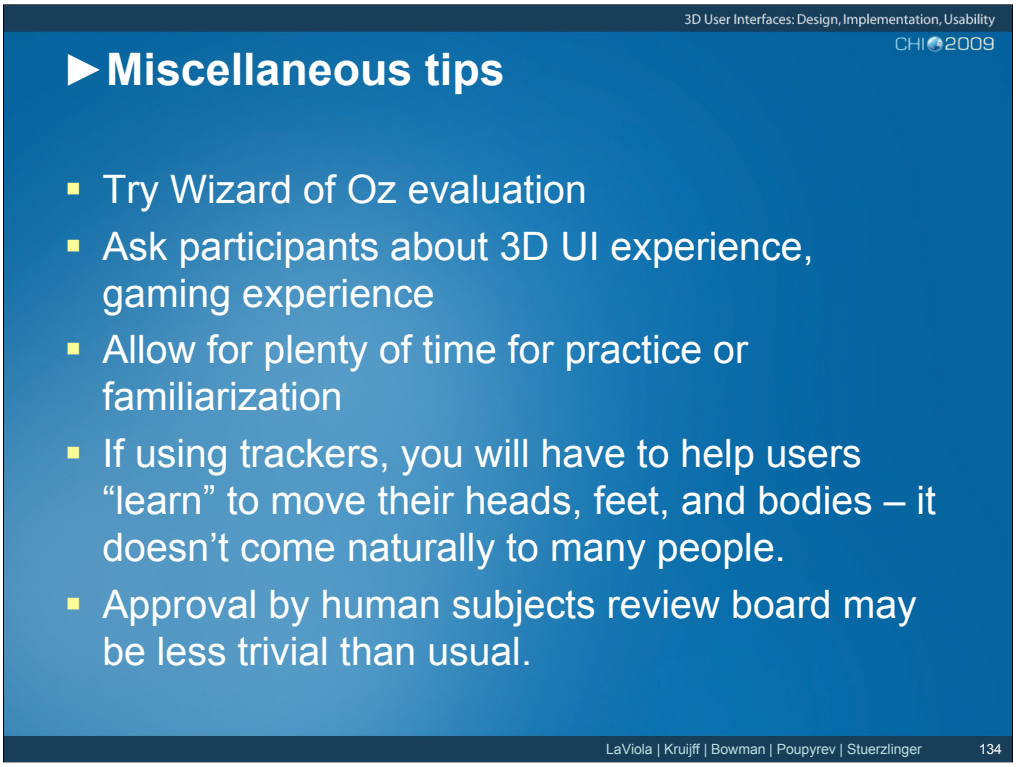
Third, we look at user issues. One problem is the lack of users who can truly be considered “experts” in 3D application usage. Since the distinction between expert and novice usage is important for interface design, this makes recruiting an appropriate subject pool difficult. Also, 3D systems have problems with simulator sickness, fatigue, etc. that are not found in traditional UIs. This means that the experimental design needs to include provisions like rest breaks and the amount of time spent in the system needs to be monitored.



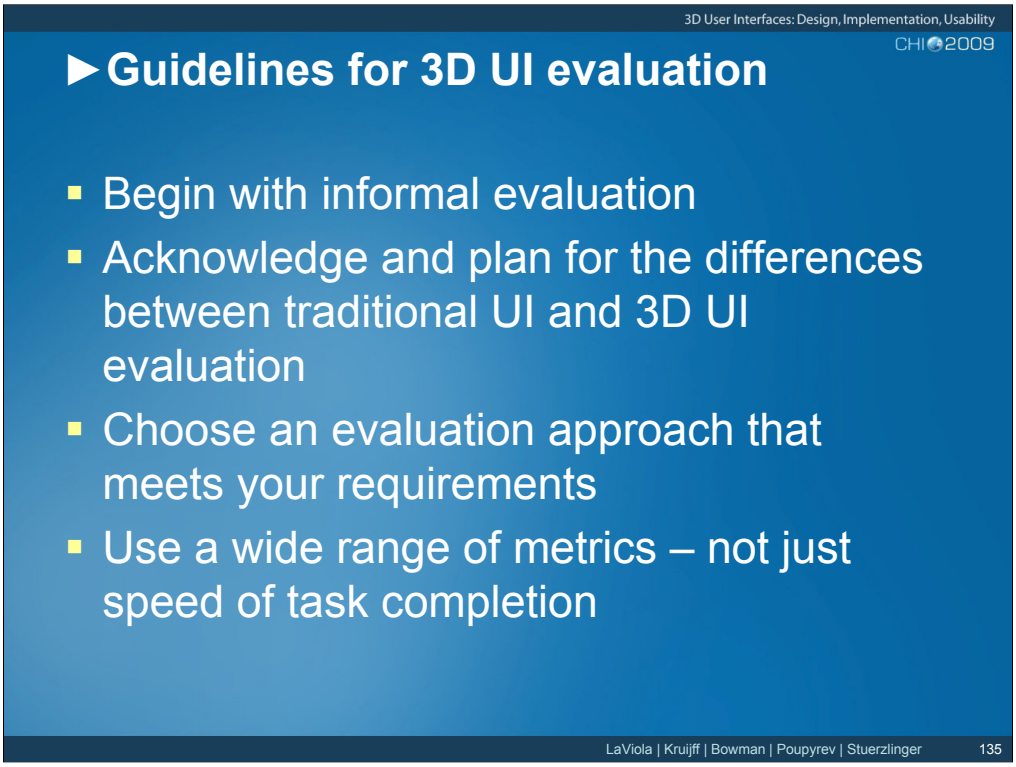
Highlighting one of the issues from the previous slide … Here are some tips on avoiding, monitoring, and dealing with sickness and discomfort issues.



Fourth, issues related to the type of evaluation performed. Heuristic evaluation can be problematic, because 3D interfaces are so new that there is not a large body of guidelines from which to draw, although this is changing. Also, if you are doing a formal experiment, there are a huge number of factors which might affect performance. For example, in a travel task, the size of the environment, the number of obstacles, the curviness of the path, the latency of the system, and the spatial ability of the user all might affect the time it takes a user to travel from one location to the other. Choosing the right variables to study is therefore a difficult problem. It's also sometimes difficult to realize which factors must be held constant to avoid excessive variability in the experimental results.



A few more important tips to consider when designing a 3D UI evaluation.



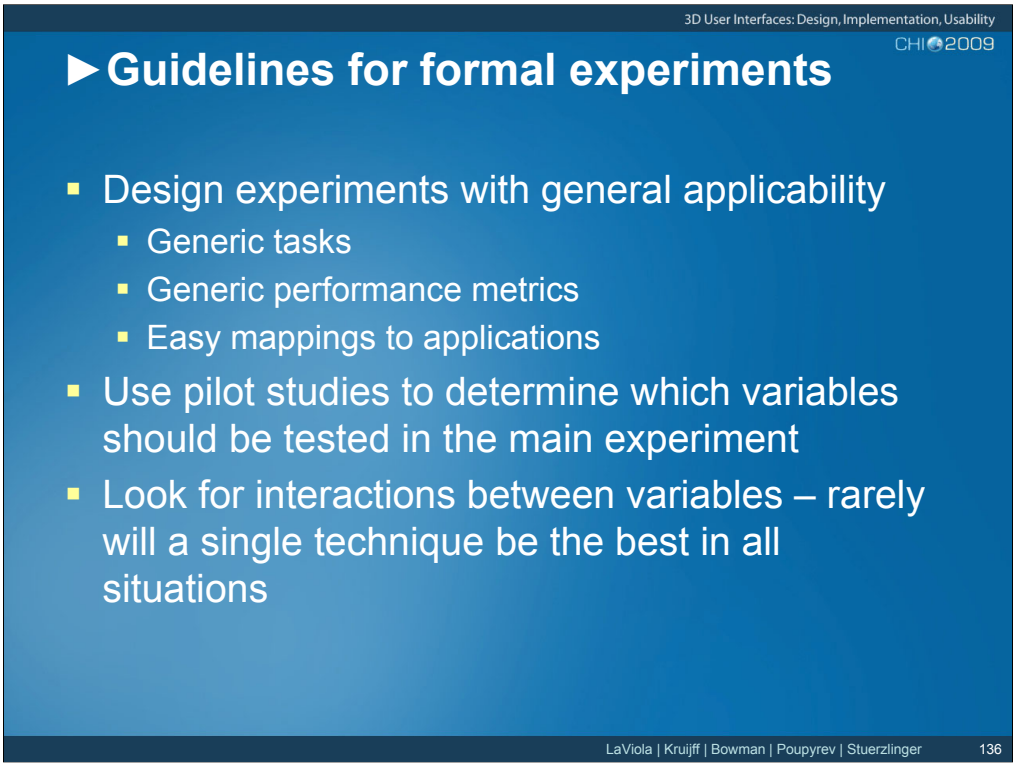
Here are a set of guidelines to be used in any type of evaluation of 3D UIs.

Informal evaluation is very important, both in the process of developing an application and in doing basic interaction research. In the context of an application, informal evaluation can quickly narrow the design space and point out major flaws in the design. In basic research, informal evaluation helps you understand the task and the techniques on an intuitive level before moving on to more formal classifications and experiments.

Remember the unique characteristics of 3D UI evaluation from the beginning of this talk when planning your studies.

There is no optimal evaluation technique. Study the classification presented in this talk and choose a technique that fits your situation.

Remember that speed and accuracy do not equal usability. Also remember to look at learning, comfort, presence, etc.

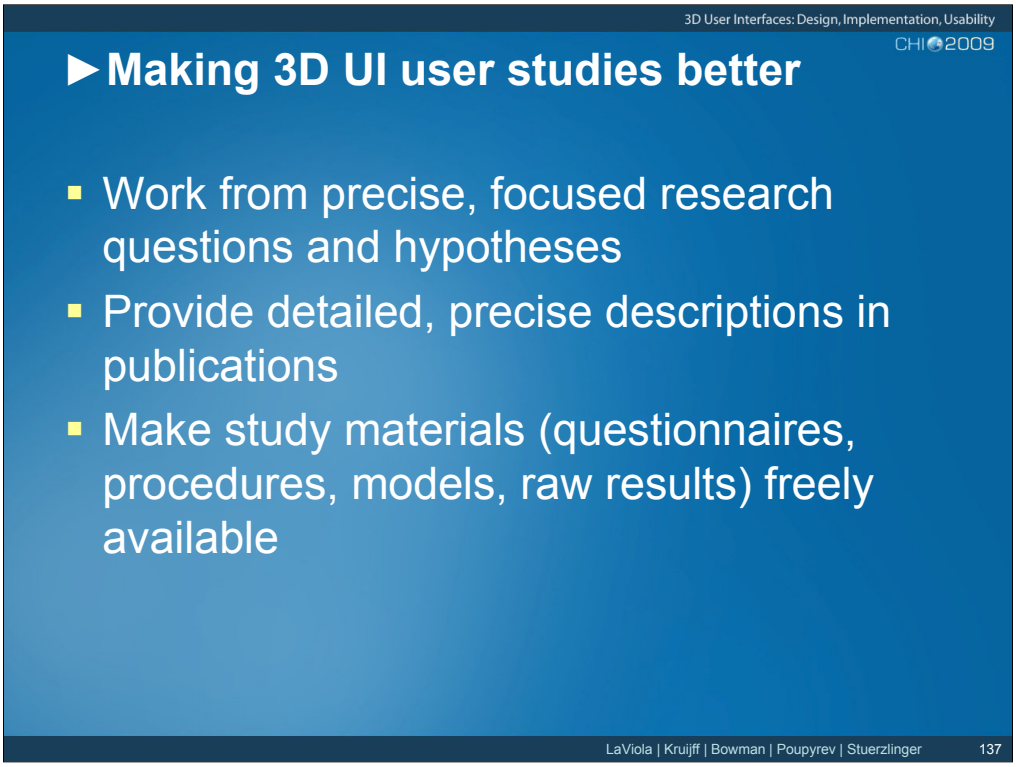


These guidelines are for formal experiments in particular – mostly of interest to researchers in the field.

If you're going to do formal experiments, you want the results to be as general as possible. Thus, you have to think hard about how to design tasks which are generic, performance measures that real applications can relate to, and a method for applications to easily re-use the results.

In doing formal experiments, especially testbed evaluations, you often have too many variables to actually test without an infinite supply of time and subjects. Small pilot studies can show trends that may allow you to remove certain variables, because they do not appear to affect the task you're doing.

In almost all of the experiments we've done, the most interesting results have been interactions. That is, it's rarely the case that technique A is always better than technique B. Rather, technique A works well when the environment has characteristic X, and technique B works well when the environment has characteristic Y. Statistical analysis should reveal these interactions between variables.



The 3D UI community is still learning how to run effective, useful user studies. Here are a few ideas for 3D UI researchers on how to make study results valid, repeatable, and useful to the entire community.

I worked closely with Joe Gabbard and Debby Hix (both of Virginia Tech) in developing the list of distinctive characteristics of 3D UI evaluation, the classification scheme for evaluation techniques, and the testbed/sequential evaluation approaches.