Predictive tracking for head-mounted displays
A dissertation proposal
Ronald Azuma
April 29, 1992
Final version

Thesis Statement: *"Predictive tracking can noticeably reduce the effects of latency in head-mounted display systems"*

0 Introduction

First demonstrated in 1968 [Suther68], head-mounted displays [HMDs] have the potential to immerse a user inside a "virtual world," where the user can view and interact with computer-generated three-dimensional environments. However, today's systems far fall short of creating completely convincing virtual environments. Much work remains in overcoming limitations in the technologies that drive HMDs: the tiny video displays placed in front of the user's eyes, the graphics engine that generates the images seen on those displays, and the tracking system that determines the position and orientation of the user's head. Equally important is a lack of demonstrated, serious applications that would greatly benefit from HMDs.

Many of us at UNC believe that some of these real applications will come from the area of *augmented reality*, in which computer-generated images displayed inside an HMD supplement, rather than supplant, the real world. We are building near-range *see-through* HMDs that let the user see the real world with computer graphics images merged in. For example, a doctor might be able to "see inside" a womb if ultrasound images of a fetus were reconstructed as a three-dimensional model seen in the doctor's HMD. Maintenance of complex machinery would be easier if three-dimensional directions projected over the equipment showed, step-by-step, how to conduct the repair work. Data taken by head-based sensors could be represented and selectively displayed to augment a user's perception of his environment. Appropriate sensors could extend the user's visual range into the infrared and ultraviolet frequencies. Remote sensors would let users view objects hidden by walls or hills.

However, making convincing augmented reality applications requires overcoming many obstacles. One of the most serious challenges is the problem of registering computer-generated objects with existing objects in the real world. As the user moves and rotates his head, the HMD system updates the computer-generated images so that the virtual objects appear as they should from the different viewpoints. If we place a virtual vase on top of a real pedestal, that vase should appear fixed to the pedestal, no matter how the user changes his vantage point.

A recent first attempt at an ultrasound application with a see-through HMD [Bajura92] vividly demonstrated that accurate registration of virtual objects to real-world objects is not easy. The virtual objects "swam" and floated in space as the user rotated his head, compromising the illusion we hope to achieve.

What are the sources of registration error? One source is the tracking subsystem. Accurate registration demands accurate head and hand tracking, and the magnetic-based

Polhemus tracker they used gives distorted readings at long ranges. Another source is distortion in the optics in the HMD. But the biggest cause of the registration problem is the end-to-end latency: the delay between the time the tracker starts measuring the head location and the time the graphics engine finishes drawing the corresponding images on the displays. If the user constantly moves his head, by the time the images get updated his head is in a different location from the one used to generate the images he sees. Thus, the virtual objects appear to "lag" behind the user's motion. The result is a virtual world that "swims about," reducing the effectiveness of the illusion and, in extreme cases, contributing to motion sickness.

It is highly unlikely that technological improvements alone will solve the latency problem. Human beings can detect astonishingly small amounts of lag. From his work on the Air Force's Super Cockpit program, Tom Furness reports that humans can perceive as little as 5 ms of lag [Furness91]. But modern graphics systems, in their quest to increase throughput, are heavily pipelined, increasing latency. For example, Pixel-Planes 5 currently requires a minimum of 66 ms to render two low-resolution images [Ellsworth92]. Tracker latencies vary widely; our optical ceiling tracker [Ward92] typically has 20-30 ms lag, whereas the Polhemus Isotrack and Ascension Bird trackers average about 55 ms and 85 ms respectively [Mine92]. And even if the graphics engine runs at 60 Hz, there is a minimum 16 ms scanout delay for the video signal alone, irrespective of the image being drawn. Since the end-to-end system latency is the sum of the all of these latencies, plus any communication delays, all reasonably powerful augmented reality systems in the foreseeable future are expected to have latencies well above the limits of human perception.

What, then, can be done to overcome latency? We should certainly continue our efforts to reduce latency in all components of the augmented reality system. Then to reduce the effects of the latency that remains, I propose developing predictive tracking techniques that estimate where the user's head will be in the future. Recall that latency causes registration problems when the user's head is in motion because the images were computed for viewpoints that lag behind the head's actual locations. Instead of feeding the raw tracker outputs to the graphics engine, we will instead use them to estimate the likely behavior of the head for short periods into the future. Then, given an estimate of how long it will take the graphics engine to generate a new set of images, we can create a predicted viewpoint and send that to the graphics engine. If our prediction is correct, the viewpoint used to generate the images the user sees will match the actual head location, resulting in no registration error due to latency.

Head-motion prediction will not be perfect and cannot work under every circumstance, because there's always some randomness associated with head motion that cannot be completely compensated for. However, I also believe that there are enough limitations and coherent behaviors in head motion that we can use to develop and demonstrate a prediction heuristic that will visibly reduce the effects of latency in head-mounted displays.

1 Previous work

Predictive tracking for HMDs is not a completely new idea, and work has also been done in related fields. We now survey these previous works.

At least three prediction methods were developed for the prototypes of CAE's Fiber Optic Helmet-Mounted Display [FOHMD]. Bernard Smith [Smith84] developed a very simple prediction method that essentially takes the last two known rotation matrices and uses the component-by-component difference to linearly extrapolate what the rotation matrix will be in the future. Neither the method nor the resulting graphs are convincing. Uwe List [List83] uses two accelerometers mounted on the helmet to detect yaw motion. The acceleration values are integrated in a very simple scheme to predict yaw orientation at a given time in the future. CAE's 1986 technical report [CAE86] describes the third and most sophisticated method. It uses accelerometers to cover all three orientation axes. These signals are integrated to predict future rotation angles, and these predictions are blended with the angles actually measured by the tracker to eliminate drift and long-term errors.

Probably the best prediction system today runs on the commercially-available FOHMD. Unfortunately, CAE considers their prediction method proprietary, so no published references are available. However, we were able to learn some facts from George Kelly [Kelly92], who worked on an FOHMD at Williams Air Force Base. That particular machine uses rate gyroscopes to measure velocity, rather than using accelerometers, but other than that the basic approach is similar to the one described in [CAE86]. The velocity values are integrated to predict future orientation angles, and these are blended with the values generated by the tracker.

At least three groups have tried Kalman filters as the basis of prediction routines. Very briefly, given a linear model of a system, a series of measurements of the state of the system, and descriptions of the noise processes that affect the system, the filter will generate an estimate of the state variables that is optimal, in the sense that the estimate minimizes the expected least-squares error. Implementing the Kalman filter is easy; the difficult part is finding the model and noise characteristics appropriate for your system.

Rebo, in his MS thesis [Rebo88], built a very simple Kalman filter for head-motion prediction that treated acceleration as white noise. He states that prediction is accurate when head motion is very slow, but also notes that when the user comes to an abrupt halt, the predicted locations oscillate.

Liang [Liang91] offers a simple but more believable Kalman filter that is based on the assumptions that head rotations are infrequent and that head translations occur mostly along the gaze direction. The Kalman filter predicts future orientation values, and a separate low-pass filter reduces jitter in the position measurements, because their Polhemus is apparently much noisier than ours are. Their graphs comparing predicted vs. actual measurements fail to label the scale used on the position axis, making it difficult to tell the size of the errors. They do note that there is a tradeoff between the smoothness of the filter output and how much it overshoots the correct values. A conversation with one of the co-authors [Green92] revealed that they are continuing this work by using a modal approach. This assumes that users orient their heads according to the following script:

1) Start rotation from a still position.
2) Rotation path follows a great arc of a circle.
3) Stop rotation.

By writing heuristics to detect these modes, they hope to do more accurate prediction and reduce some of the overshoot problems.

While Liang only predicts orientation, Friedmann [Friedmann92] uses Kalman filters for predicting the position of a drumstick. With a Polhemus sensor taped to the drumstick, he tries to detect when the drumstick strikes a surface so he can play a computer-generated sound. Latency in the tracker forces him to predict when the collision occurs. His physical model is simple and the noise model is basically a set of educated guesses. Recently he has moved to a Multiple Model or Generalized

Likelihood approach, which tries to reduce prediction error by running several different Kalman filters in parallel and then choosing the output of the one that seems to best match the observed measurements. His graphs indicate that the predicted output overshoots by tens of millimeters, but since all his application wants is the instant when the collision occurs, overshoots are not nearly as big a problem as they are for HMD motion.

Richard So [So91] uses a technique called image deflection, in which the graphics engine rasterizes an image much larger than needed to fill the HMD display. Then, just before scanning the image out, a very simple prediction routine guesses what the actual orientation is and uses that to select what fraction of the frame buffer to send to the display. Unfortunately, this ingenious trick works only for orientation, and our existing Pixel-Planes 5 frame buffers are not capable of supporting this technique. It's main advantage is reducing how far into the future a prediction heuristic needs to estimate future locations.

Wang [Wang90] from IBM Yorktown Heights mentions, but does not describe, a prediction system for tracking hand motion. They have had problems with overshoots caused by rapid acceleration or deceleration, and a conversation with Dan Ling indicated that they have not done much work in this area.

The flight simulation world has also attempted to deal with latency problems; examples include Crane [Crane84] and McFarland [McFarland86] [McFarland88]. These do not directly apply to HMDs or head motion. Instead, they model the aircraft and the pilot controlling it in the simulator, attempting to increase the stability and reduce the apparent lag of the system. McFarland's papers, for example, focus on the roll rate of the XV-15 aircraft. Still, this literature can serve as a source of ideas.

A wealth of literature exists on tracking and estimation problems, primarily for military applications, such as how to track the path of an enemy airplane, or how to predict the path of an incoming ballistic missile. Representative examples include [Blom84] [Chang84] [Berg83] [Bolger87] [Tugnait87]. Because the range and pattern of motions of the human head are significantly different from those of aircraft, ships, and missiles, the routines developed for target tracking and flight simulation cannot be directly applied to HMDs, but they can serve as inspiration for more sophisticated methods of head-motion prediction.

2 Contribution

Despite the previous work in this field, it appears that sufficient "daylight" remains for a dissertation in this area.

First, more work needs to be done in exploring and developing better prediction heuristics. The advent of near-range, see-through HMDs demands this. Latency effects become much more pronounced when the user can see real-world objects with little or no lag, while simultaneously viewing virtual objects with large amounts of lag. Virtually no work has been done on prediction for near-range, see-through HMDs. Existing techniques tend to suffer from large overshoots and ringing, focus on only orientation or position, and make many assumptions about head motion that are not backed up by collected data. I believe that a more comprehensive attack on this problem has a good chance to improve upon existing prediction heuristics.

How should we evaluate prediction methods? CAE justifies its existing method by saying "pilots like it better than no prediction." While understandable, this approach is highly subjective and offers no quantifiable measure of performance. Most other papers evaluate their methods by showing a graph of the predicted signal vs. the original signal. These are helpful, but they only show the effects along one dimension and are not easily translatable into what really counts: what kind of error shows up on the user's displays? Evaluation in previous work emphasized what "feels better," but in our upcoming see-through HMDs, evaluation should focus on the accuracy of the prediction and on measuring how much registration error occurs. We need to define performance metrics that provide a quantifiable basis of comparison and be able to justify these metrics from a perceptual standpoint.

Finally, when existing papers describe their prediction methods, they rarely tell you how they determined the parameters that control the method. CAE provides a cluster of physical knobs that set the coefficients in their prediction filters. The user adjusts the knobs until the system "feels right." Most others pick parameters by experimentation or other completely empirical methods. But when I try to implement these routines on my HMD system, do I simply plug in their constants, when my head unit has different inertia characteristics, and the tasks demanded in my HMD applications may result in head motions that violate their assumptions? This field requires prediction methods with built-in procedures that adapt them to different HMDs and different applications.

I offer the following as new contributions from a thesis in this area:

1) Design, implementation, and demonstration of a prediction method that performs better than existing methods
2) Defining quantifiable metrics to evaluate prediction methods
3) Developing procedures that allow my prediction technique to be moved to other HMDs, trackers, and applications

3 Approach and Evaluation

1) Design, implementation, and demonstration of a prediction method that performs better than existing methods

   I do not have a specific method developed yet, so I will instead describe why there is room for improvement in existing prediction methods and what approaches I plan on taking.

   If the human head could move to any position and orientation instantaneously and typical head motion exhibited no correlation between adjacent time intervals, then the prediction problem would be hopeless. If head motion was completely determined from a set of initial conditions, then the prediction problem would be trivial. The actual situation lies somewhere in between, which makes the problem interesting.

   A person wearing an HMD has limits on how fast he can accelerate and decelerate his head. The inertia of his head and the head unit itself, combined with limited muscle strength, causes these limits. I recorded my head motions underneath the ceiling tracker when I tried to walk around and rotate my head as fast as I could without losing my balance or running into walls. Gary Bishop converted this data into the Fourier frequency domain and observed that almost all of the energy lies in frequencies of 2 Hz or less. This matches Richard So's statement that the bandwidth of human head motions is 0 to 1.5 Hz [So91], and the shape of our frequency response curve looks very similar to Liang's [Liang91]. The limited bandwidth of head motion puts quantifiable limits on how quickly head motion can change.

Furthermore, when I looked at graphs of my rapid head motion, I found that the resulting curves were usually quite smooth, up to the noise limits of the optical tracker itself. If even rapid motions produce smooth, well-behaved curves for outputs, then we will have decent signals to base prediction upon.

If prediction is feasible, the next question is: why should I be able to do better than previous attempts? Basically, I intend to do a better job of characterizing and modeling head motion in HMD systems, drawing knowledge from related fields to develop something more sophisticated than existing routines. That, combined with careful implementation and access to more advanced hardware, should give me a reasonable chance of success. Previous work in this area relies too heavily on assumptions and educated guesses; we need to do more data gathering and analysis to improve upon existing techniques.

The first step, already underway, is to build a testbed in which we can try out various prediction routines. The testbed consists of the ceiling tracker directly connected to Pixel-Planes 5, avoiding the uncontrollable delays caused by UNIX, and providing accurate measurements of the latencies in each system component. A graphics processor (GP) will be freed to receive the tracker information, run prediction heuristics, and then send the predicted viewpoints to the Master GP for rendering. Without this control over time, it's difficult to have confidence in any prediction technique, because it wouldn't know how far into the future to predict!

Once the testbed is in place, we can test existing prediction heuristics on our ceiling HMD, tweaking the parameters to make them fit our system. I do not expect any of them to do well. In particular, I will work to implement Liang's method as best I can on the ceiling tracker. As the best published method, this will become the baseline for comparison, the routine that I must improve upon.

Additional hardware is needed to do better. Accelerometers or rate sensors will be vital to the success of any method. Without them, a tracker 1) cannot report velocity or acceleration changes until after they occur, and 2) must estimate velocity and acceleration values by doing numeric differentiation, an inherently noisy process. With the help of the MSL, I intend to acquire and attach accelerometers and/or rate gyroscopes to one or more of our head trackers. To simplify the problem, I will initially focus on orientation rate sensors.

Better characterization of noise parameters and head motion will also improve prediction. This will require a substantial amount of data gathering and statistical analysis. Existing techniques determine these by taking educated guesses or making assumptions; a more reliable way is to take measurements. This may also let us build more accurate models. For example, my first goal will probably be building a more accurate Kalman filter, where the noise parameters have been measured statistically and the velocity values come from rate gyroscopes. Instead of modeling acceleration as white noise, as existing methods do, we'll characterize it by a table of Bayesian prior probabilities, based on measured head motion. Different datasets would "tune" the predictor to different applications or users.

Other correlations that aren't readily apparent might be discovered by running collected datasets through a neural network.

It's also important to work on reducing the existing latency in our tracking and graphics engine subsystems. The farther into the future a method has to predict, the more severe the error will be, so faster systems will give us greater chances of success. The ceiling tracker is reasonably fast, with an average 20-30 ms latency, but it could be improved, and the Fastrak promises to have even less lag. Others are already working on reducing the latency in Pixel-Planes 5 for certain applications. Image deflection would be a great help in reducing the prediction distance, if we can find a graphics engine that can support it.

The final, improved prediction routine will probably be a variation on or a combination of existing advanced techniques, such as multiple Kalman filters running in parallel or a series of filters resembling CAE's technique, and I expect it will be inspired by techniques from the tracking and estimation literature and from control theory. I don't expect to find a completely new method that will supersede Kalman filters, PID filters, etc. The key to improvement is not in the method per se, but in the parameters that control the method and in the data fed into the filters. These we can improve by accurate system modeling, careful implementation, and system improvements, such as adding rate gyroscopes or accelerometers.

2) Defining quantifiable metrics to evaluate prediction methods

We should convert the ceiling HMD into a see-through HMD. That will make the latency more apparent and make it easier to evaluate and compare different prediction techniques. It will also make the optical distortion more apparent. To remove that factor, we may have to make the graphics engine predistort the computed image to compensate for the optical distortion.

In near-range see-through HMDs, it's important that the virtual objects appear registered to the real world, so one possible first metric measures how much screen-space error occurs in drawing a hypothetical object placed at a fixed distance away from the HMD. This computes the error in terms of what the user can see, such as the apparent vertical and horizontal error in screen-space, measured in pixels.

However, measuring the magnitude of these errors alone is not sufficient. The smoothness of the resulting path is also an important factor. My experience with "beacon switching error" in developing the ceiling HMD tells me that if the predicted path appears ragged and discontinuous, users will not consider it satisfactory, even if the average apparent latency is less than with no prediction at all.

Furthermore, users may not be as sensitive to some aspects of latency as others. For example, when the user rapidly moves his head, he may be less sensitive to registration errors than when keeping his head completely still. This suggests some sort of weighting based on velocity. CAE takes the extreme view that errors while the head is in motion are unimportant. I don't believe that will be true in our situation, but the correct answer is to try it and find out.

How do we justify our choice of metrics? It's trivial to define an arbitrary function to provide numeric output, given numeric input. We must be able to argue that the metrics are reasonable, and the ultimate basis is perception. The strongest support would come from a series of comprehensive user studies, a task probably worthy of a PhD by itself. Instead of pursuing *findings* [Brooks88], I propose to justify my metrics through *rules-of-thumb* and *observations*, generated by small user studies based on a few tasks that users must perform. These tasks should cover a wide spectrum of ranges, scales, speeds, and precision in user motion. The user studies must be designed to compensate for training effects and diversity in user skills, but not to the degree that findings would require. Videotaping the subjects, with synchronized recording of tracker data, will aid documentation and analysis.

What kinds of observations and rules-of-thumb do we hope to see? I would like to discover if people are more sensitive to certain aspects of latency than others, since that will help us design prediction methods. Also, tying the perception results to the parameters that control the prediction method would be useful. For example, if user studies indicate that task K results in user motion below Q Hz, then explain how to adjust the prediction parameters to that situation.

Once the metrics are in place, how do we actually use them to evaluate prediction methods? We have a simulator for the ceiling tracker on which we can run a prediction heuristic on a simulated or actual path and determine how well it does, according to an error metric. With a see-through HMD, we can also evaluate methods at runtime by superimposing a virtual ruler near a real-world ruler. As the user moves and rotates his head, lag will cause offsets between the virtual and real ruler that the user can observe. Buttons will let the user turn prediction on and off or switch between different prediction methods, allowing direct comparisons. By itself, this runtime comparison will probably not be sufficient, because the evaluation method will rely on more than just the magnitude of the offsets. However, it’s a place to start.

Evaluation will also include an exploration of the conditions when the method works and when it fails, and will describe the sources of error and their effects on the performance.

3) Developing procedures that allow my prediction technique to be moved to other HMDs, trackers, and applications

The procedure should fall out of the determination of the parameters that control the prediction routine, since I intend to base them on measurable traits. This also means that a large amount of work may be involved in reproducing the predictor, since it will probably require a thorough understanding of the system. I’m not claiming that it will be easy, just that it should be possible.

How do we test the procedure? A sufficient demonstration is to move the method to *one* other system. For example, if the method originates on Pixel-Planes 5 and the ceiling tracker, a good demonstration would be to move it to a Fastrak on a different HMD, driven by an SGI graphics engine. If that is not possible, then another route is to modify the behavior of the ceiling tracker, such as making it slower and less accurate, and then demonstrate that the procedure can adapt the predictor to the “new” tracker.

4 Deliverables

My thesis work should deliver the following:

1) An implementation and demonstration of a prediction technique for HMDs that performs better than the best published method.

2) Quantifiable metrics for evaluating and comparing prediction methods for HMDs, justified by small user studies.

3) A procedure for implementing this routine on different HMD systems.

My committee has ruled that deliverable #3 can be considered expendable, if need be.

5 Estimated Time Table

1st summer session 92
- Teaching COMP 96 "Computers and Society" to fulfill PhD teaching requirement
- Find and order gyroscopes/accelerometers

2nd summer session 92
- Prediction testbed in place
- Try existing prediction techniques on ceiling HMD
- Begin data gathering and statistical analysis
- Begin orals preparation

Fall 92
- Orals
- Ceiling HMD converted to see-through
- Gyroscopes/accelerometers arrive and made operational
- Start designing user studies
- Develop quantifiable metrics
- Explore advanced prediction techniques

Spring & Summer 93
- Evaluate improved prediction routine
- User studies
- Demonstration of portability
- Finish dissertation writing
- Defend

References

[Bajura92] Bajura, Mike, Henry Fuchs, Ryutarou Ohbuchi. Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery within the Patient. UNC Chapel Hill technical report TR 92-005 [accepted to SIGGRAPH '92].

[Berg83] Berg, Russell F. Estimation and Prediction for Maneuvering Target Trajectories. IEEE Trans. on Automatic Control, AC-28, 3 (March 1983) pp. 294-304.

[Blom84] Blom, H. A. P. An Efficient Filter for Abruptly Changing Systems. Proceedings of 23rd Conference on Decision and Control (Las Vegas, NV, Dec. 1984) pp. 656-658.

[Bolger87] Bolger, P. L. Tracking a Maneuvering Target Using Input Estimation. IEEE Trans. on Aerospace and Electronic Systems, AES-23, 3 (May 1987) pp. 298-310.

[Brooks88] Brooks Jr., Frederick P. Grasping Reality Through Illusion -- Interactive Graphics Serving Science. Proceedings of SIGCHI '88 (Washington D.C., 15-19 May 1988) pp. 1-11.

[CAE86] Flight simulator wide field-of-view helmet-mounted infinity display system. Technical report AFHRL-85-59, Williams AFB, AZ: Operations Training Division Air Force Human Resources Laboratory (May 1986), pp. 48-64.

[Chang84] Chang, Chaw-Bing, John Tabaczynski. Application of State Estimation to Target Tracking. IEEE Trans. on Automatic Control, AC-29, 2 (Feb. 1984) pp. 98-109.

[Crane84] Crane, D. F. The effects of time delay in man-machine control systems: implications for design of flight simulator visual-display-delay compensation. Proceedings of the IMAGE III conference (1984) pp. 331-343.

[Ellsworth92] Ellsworth, David. Personal communication, Feb. 17, 1992.

[Friedmann92] Friedmann, Martin, Thad Starner, and Alex Pentland. Device Synchronization Using an Optimal Linear Filter. Proceedings of 1992 Symposium on Interactive 3D Graphics (Cambridge, MA, 29 March - 1 April 1992) pp. 57-62.

[Furness91] Furness, Thomas. Personal communication, Aug. 1, 1991.

[Green92] Green, Mark. Personal communication, March 24, 1992.

[Kelly92] Kelly, George. Personal communication, Feb. 7, 1992.

[Liang91] Liang, Jiandong, Chris Shaw, Mark Green. On Temporal-Spatial Realism in the Virtual Reality Environment. Proceedings of the 4th Annual ACM Symposium on User Interface Software & Technology (Hilton Head, SC, Nov. 11-13 1991) pp. 19-25.

[List83] List, Uwe H. Nonlinear Prediction of Head Movements for Helmet-Mounted Displays. Technical report AFHRL-TP-83-45, William AFB, AZ: Operations Training Division Air Force Human Resources Laboratory (Dec. 1983).

[McFarland86] McFarland, Richard E. CGI Delay Compensation. NASA Technical Memorandum S6703 [N88-12932] (1986).

[McFarland88] McFarland, Richard E. Transport Delay Compensation for Computer-Generated Imagery Systems. NASA Technical Memorandum 100084 [N88-17645] (Jan. 1988).

[Mine92] Mine, Mark. Personal communication, March 1992.

[Rebo88] Rebo, Robert. A Helmet-Mounted Virtual Environment Display System. MS Thesis, Air Force Institute of Technology (Dec. 1988).

[Smith84] Smith, Bernard R. Jr. Digital Head Tracking and Position Prediction for Helmet-Mounted Visual Display Systems. AIAA 22nd Aerospace Sciences Meeting (Reno, NV, Jan. 84).

[So91] So, Richard H. and Michael Griffin. Effects of Time Delays on Head Tracking Performance and the Benefits of Lag Compensation by Image Deflection. AIAA Flight Simulation Technologies Conference (New Orleans, LA, Aug. 12-14 1991) pp. 124-130.

[Suther68] Sutherland, Ivan. A head-mounted three dimensional display. Fall Joint Computer Conference, AFIPS Conference Proceedings, 33 (1968) pp. 757-764.

[Tugnait82] Tugnait, Jitendra K. Detection and Estimation for Abruptly Changing Systems. Automatica, 18, 5 (Sept. 1982) pp. 607-615.

[Wang90] Wang, Chu P., Lawrence Koved, Semyon Dukach. Design for Interactive Performance in A Virtual Laboratory. Proceedings of 1990 Symposium on Interactive 3D Graphics (Snowbird, UT, 1990). In *Computer Graphics*, 24, 2 (March 1990) pp. 39-40.

[Ward92] Ward, Mark, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs. A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems. Proceedings of 1992 Symposium on Interactive 3D Graphics (Cambridge, MA, 29 March - 1 April 1992) pp. 43-52.