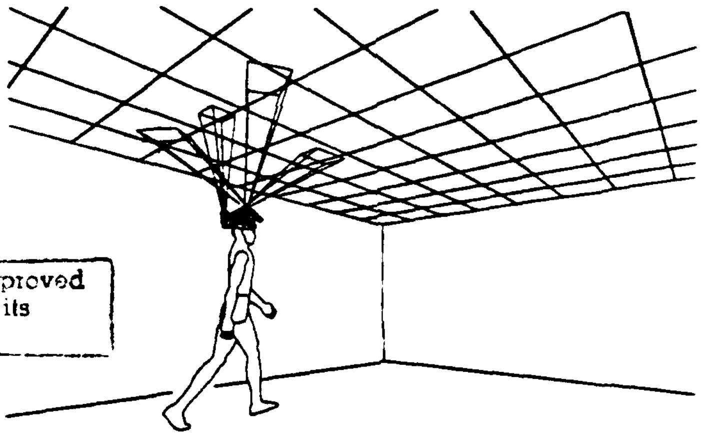
1

# Advanced Technology for Portable Personal Visualization

## Report of Research Progress
## April – December 1991

This research is supported in part by
DARPA ISTO Contract No. DAEA 18-90-C-0044

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175
Telephones: 919-962-1911 (Fuchs), 919-962-1931 (Brooks)
FAX: 919-962-1799
E-mail: fuchs@cs.unc.edu, brooks@cs.unc.edu



*Principal Investigators*: Frederick P. Brooks, Jr., and Henry Fuchs

*Faculty*: Steve Pizer, Kenan Professor; Vern Chi, Director, Microelectronic Systems Lab; John Eyles, Research Assistant Professor; Gary Bishop, Research Associate Professor; Doug Holmgren, Visiting Assistant Professor

*Staff*: Warren Robinett (Senior Researcher), Jannick Rolland (Optical Engineer), David Harrison (Hardware Specialist, Video), John Hughes (Hardware Specialist, Force Feedback ARM), Linda Houseman (Administration), Kathy Tesh (Secretary), Fay Ward (Secretary)

*Research Assistants*: John Alspaugh (Architectural Walkthrough), Ron Azuma (Optical Tracker), Jim Chung (Team Leader, Graphics Software), Drew Davidson (Sound User Interface), Erik Erikson (Molecular Graphics), Stefan Gottschalk (Tracker Simulation Software), Rich Holloway (Graphics Library), Phil Jacobsen (Optical Self-Tracker), Mark Mine (Bathysphere), Uwe Nimscheck (Visiting RA, Architectural Walkthrough), Russell Taylor (Communication Library), Amitabh Varshney (Architectural Walkthrough), Yulan Wang (Architectural Walkthrough), Hans Weber (Architectural Walkthrough)

# Advanced Technology for Portable Personal Visualization

## 1.0 Summary

Despite the recent avalanche of publicity on Virtual Reality, the state of the art in head-mounted displays is still at the "toy" stage. Although extravagant claims are made almost daily on TV and in the popular press, there is very little that can be usefully accomplished today with Virtual Reality technologies.

We ~~at UNC~~ have been working since the early 1970's on aspects of these technologies and have been advancing the state of the art by a "driving problem" approach; we let the needs of selected applications stimulate the direction of the technological developments, and then test new results by their impact on solving the original application. We have been working on three application areas: molecular modeling, 3D medical imaging, and modeling of architectural interiors.

We have put together complete systems that include the head-mounted display device, the display generation hardware, and the head and hand tracker. We bought the components when available and built the components when there was a clear advantage to doing so:

a) We built, under separate major funding by DARPA and NSF, the display generation hardware (our most recent machine is Pixel-Planes 5),

b) We built an electro-optical helmet-tracking system (the "ceiling tracker") that can determine position and orientation over a large area (currently under a 10 x 12 foot ceiling) — to our knowledge, the first demonstrated scalable-area helmet tracker.

c) We buy head-mounted display devices, but continue to build head-mounted display devices with "see-through" capability -- the ability for the wearer to observe virtual world objects superimposed onto his/her physical surroundings.

Our major accomplishment in this period was a week of comprehensive demonstrations at SIGGRAPH'91, the world's premier computer graphics conference (Las Vegas, July 28 - August 2, 1991). In the juried Tomorrow's Realities Gallery, the UNC group was allocated the largest area of any exhibitor (28 x 42 feet), and was approved for the largest number of distinct demonstrations (7):

1) Moving about the interior of a virtual building while walking under our custom 10 x 12-foot ceiling,
2) Using a treadmill with steerable bicycle handlebars to walk through larger areas of a virtual building,
3) "Flying" through a space filled with various models of large molecules,
4) Bicycling through an outdoor environment on a bike whose pedals exert forces depending on the steepness of the terrain,
5) "Flying" about an outdoor scene that included realistically textured clouds and terrain,
6) Designing the configuration of multiple radiation therapy beams to focus onto a tumor nestled inside a patient model with multiple (healthy) major organs,
7) "Flying" around a virtual world in which a second, independently moving, user is modeling 3D objects (feels like being immersed in a 3D MacDraw world).

All these demonstrations were for individual users; the guest wore a stereo, tracked head gear and interacted with the system for about 4 minutes. Crowds around the booth area observed on overhead monitors the images that the individual saw in the head gear.

## 1.1 Goals of the Head-Mounted Display Project

- Demonstrate the usefulness of the head-mounted display (HMD) in real applications.
- Improve the hardware subsystems which currently limit performance of HMD (tracker, HMD optics and display, real-time graphics generation).
- Design and implement software base to support HMD functions.
- Integrate visual, auditory and haptic (force feedback) displays into a working system.
- Build new input devices and investigate methods of manual control suitable to a head-mounted display.

## 1.2 Goals of the Optical Tracker Subproject

- Develop long-range trackers for head-mounted displays. For the short term, develop a system that can track an HMD inside a room-sized environment. For the long term, explore technologies that have unlimited range without requiring modification of the environment.
- Work on improving the other aspects of HMD tracking: latency, speed, accuracy, and resolution.

## 2.0 Summary of Major Accomplishments

- The UNC systems (including the ceiling tracker) were successfully demonstrated with seven distinct applications in the Tomorrow's Realities Gallery at the SIGGRAPH '91 conference in Las Vegas.

## 2.1 Head-Mounted Display System

- We demonstrated a prototype system integrating ultrasound scanner input into a virtual world in the Tomorrow's Realities Gallery at the SIGGRAPH '91 conference.
- We demonstrated a prototype system integrating monocular live video with opaque HMD in the Tomorrow's Realities Gallery at the SIGGRAPH '91 conference..
- We have ported all major HMD applications from Pixel-Planes 4 to Pixel-Planes 5.
- The main libraries for virtual-worlds applications now support applications on either host with minimal changes.
- Adlib, a library for handling analog-digital devices in a transparent manner has been created and fully documented.
- Soundlib, a library for playing sounds on the Macintosh has been created and documented. Also, we acquired a library of more than 1,000 sounds for use with the sound Macintosh.
- All HMD-based programs now use the new base software with vlib, trackerlib, and adlib.
- Sdilib, a serial-device interface library, has been created and documented. Sdilib allows applications to communicate with serial devices across the network as if they were connected to the user's machine.

- Wè have created many new virtual-worlds applications. Among them are:
  - 3dm: a three-dimensional modeler. This application grew out of a software engineering class project and is now a fully documented tool for creating three-dimensional models from *within* the virtual world.
  - Vixen: a model-viewing program. This application is also fully documented as a general-purpose tool and allows the user to view any 3D model, grab it, scale it, and fly around it.
  - Gallery: a tool for creating and exploring interactive environments rapidly and easily. It can be used to do rapid prototyping of interaction methods and world layout or to create interesting worlds to explore.
  - Rtpdemo: a program for demonstrating the placement of radiation beams for cancer therapy. Related to James Chung's dissertation work.
  - Viper: an adaptation of *vixen* designed to deliver near-real-time performance. This application monitors its update rate and simplifies the scene until the requested rate is achieved. Users can press a button to request a full-detail view of the scene once they have found the area in which they are interested.
- We integrated the new ceiling tracker and associated HMD into the system; any existing application can run under the UNC ceiling tracker as if it were just another tracker/HMD combination.
- We integrated the force-feedback ARM into trackerlib so that it can be used as a tracker.
- All virtual-worlds applications can now use sound, visuals, and force feedback.
- We have done and evaluated user studies to explore different modes of interaction with HMDs and other devices on Pixel-Planes 4.
- The pre-prototype "bathysphere" system is working. The "bathysphere" uses head-tracked stereo to achieve an out-the-window view of a virtual world.
- We built and put in use new hand-held input devices.

## 2.2 See-through Head-Mounted Display with Custom Optics

- We have completed the optical design of a 30-degree field-of-view system using off-the-shelf lenses.
- We assembled and tested the 30-degree field-of-view system on an optical bench in our optical lab.
- The diverse optical elements are currently in the mechanical shop being assembled into a HMD.
- We completed the optical design of a 60-degree field-of-view system using custom optics.

## 2.3 Tracking

- We constructed a full-scale optical tracking system in time to be demonstrated to the public at the ACM SIGGRAPH '91 conference (28 July–2 August 1991). The system features a scalable work area that currently measures 10' x 12', a measurement update rate of 20–100 Hz with 20–60 ms of delay, and a resolution specification of 2 mm and 0.2 degrees. The sensors consist of four head-mounted imaging devices that view infrared light-emitting diodes (LEDs) mounted in a 10' x 12' grid of modular 2' x 2' suspended ceiling panels. Photogrammetric techniques allow the head's location to be expressed as a function of the known LED positions and their projected images on the sensors. The work area is scaled by simply adding panels to the ceiling's grid. Discontinuities that occurred when changing working sets of LEDs were reduced by carefully managing all error sources, including LED placement tolerances, and by adopting an overdetermined mathematical model for the computation of head position: space resection by collinearity. To our knowledge, this is the first demonstrated scalable tracking system for HMDs.

- After its return from SIGGRAPH '91, the optical tracker was installed in our graphics laboratory, where it runs today. Further improvements to the system since then include:
  - Doing hand-tracking by installing a Polhemus source on the head unit and a sensor on the hand. The limited range of the Polhemus is not a problem here because it is physically impossible to move your hand out of range and magnetic trackers are not subject to line-of-sight restrictions that make optical hand tracking a difficult problem. This demonstrates a *hybrid* tracker system, using different technologies to cover each other's weaknesses. The hand controller, fabricated in-house, is a modified bike glove with two buttons for input.
  - Improving the heuristics for choosing which LEDs to light, increasing the average update rate and reducing the average latency.
  - Finding and acquiring a set of smaller, lighter lenses. We will attempt recalibration with these new lenses. If they work, they will remove more than 2 lbs. of weight from the head.
  - Recording capabilities have been added (timing and position), which helps us collect data for future tracking studies.
  - The simulator for this tracker has been updated to reflect its current configuration and capabilities.
- We wrote a conference paper and a technical report describing the optical tracker (see Appendix D).
- We have begun informal, interview-based user studies of the impact of the optical tracker on the design of a kitchen . This is being done in conjunction with the UNC Walkthrough group.
- In conjunction with Dr. John F. Hughes of Brown University, we are developing calibration techniques for reducing systematic error sources in the optical tracker.
- We have begun studies exploring the feasibility of predictive tracking (library research, initial experiments with Kalman filters, data collection from our optical tracker).
- Efforts to build a head-tracked stereo display system are currently focusing on overcoming problems with the stereoscopic display. In the interim, tracking is being done by a magnetic-based system (the Bird), but we reserve the option to replace that with an inside-looking optical system, using hardware similar to that used in the existing optical tracker.
- We fabricated prototype Self-Tracker sensors through MOSIS.
- We completed the interface control board for the cluster of Self-Tracker chips.
- The design for the software simulation of Self-Tracker chips using Pixel-Planes 5 is in progress.

## 2.4 Interactive Building Walkthrough

- The Walkthrough system (both the treadmill and ceiling tracker versions) was successfully demonstrated in the Tomorrow's Realities Gallery at the ACM SIGGRAPH '91 conference in Las Vegas.
- We have added a capability to display models in real-time on Pixel-Planes 5 as they are being built and modified on the Macintosh using the Virtus WalkThrough program.
- We developed textures to increase the model realism, including a flickering fire texture [Rhoades92], with appropriate lighting effects.
- We added location-dependent, monaural sound cues to the system.
- We have begun running user studies with Walkthrough under the ceiling tracker to study the effectiveness and shortcomings of the existing system.
- We have installed in pilot versions of our code the ability to "open" doors in the model.

# 3.0 Expected Milestones during the Next 12 Months

## 3.1 Head-Mounted Display System

- Integrate Vlib (which is polygonally based) with the volume rendering work on Pixel-Planes 5.
- Integrate ultrasound input more uniformly into a virtual world.
- Build and integrate an interim 30-degree, see-through HMD.
- Conduct basic perceptual experiments on what sorts of virtual objects can be successfully superimposed on the real world with a see-through HMD.
- Experiment with new types of hand-held input devices.
- Have the multiscreen bathysphere working and integrated with vlib.
- Perform a user study to explore the advantages of head-tracked steering in targeting of radiation therapy treatment beams.
- Calculate tracker delay compensation using predictive algorithms for the HMD system.
- Be able to explore and modify molecular surfaces in real time with force feedback and an HMD.

## 3.2 Optics

- Set up another bench prototype of the 30-degree field-of-view system using the same optics and robust and adjustable mechanical structures to carry out perceptual studies.
- Build the first working prototype 60-degree, see-through HMD system.
    - Mock-up the effect of a 10mm pupil for a 60-degree field-of-view optical system in the optical lab.
    - Assemble and test the 60-degree field-of-view system on the optical bench.
    - Head mount the 60-degree field-of-view system.

### 3.3 Tracking

- Continue incremental improvements in the optical tracker. Desired goals include:
  - Reducing the weight of the head unit, first with light lenses and later by designing a new head unit with integral photodiodes and lenses to reduce the weight further and provide greater range of head motion.
  - Increasing the ceiling structure to around 20' by 20' in order to provide much greater range of motion, both quantitatively and psychologically.
  - Investigating the feasibility of making the system wireless. This tracker actually encourages users to walk around a large area, running the risk of tripping over the supporting cables.
  - Optimizing low-level code to increase performance.
- Continued exploration of calibration techniques that will make cellular tracking systems easier to construct and have better performance.
- Develop and demonstrate predictive tracking techniques for HMDs. Learn what is required in the system to support such techniques and discover what limitations they have.
- Exploring technologies for unlimited range tracking in unstructured environments. We know of two technologies that have this potential: Self-Tracker, which we are developing, and inertial technologies. Both are *relative-mode* trackers that measure only the relative differences in position and orientation as the user moves, integrating these differences over time to recover the head's location. The main problem with these technologies is drift, since repeated integration accumulates error over time. Initial systems will probably be hybrids with our optical tracker, using it to provide auxiliary information to evaluate the technology and help it overcome drift by occasionally providing it with a good fix of the true location.
- Continue Self-Tracker development.
  - Develop Self-Tracker report processing algorithms.
  - Complete full software simulation of Self-Tracker chips.
  - Integrate Self-Tracker chips with board-level interface.

### 3.4 Interactive Building Walkthrough

- Come up with an interactive model design, with one participant using the UNC Ceiling Tracker to make design decisions, and another using the Virtus WalkThrough modelling program to carry them out.
- Add real-time, interactive radiosity to the display program on Pixel-Planes 5.
- Add real-time model mesh-generation to the display program. This is under development by a visiting student as part of his master's work.
- Conduct additional user studies.

# 4.0 Discussion of Research

## 4.1 Head-Mounted Display System

### *Hardware*

The current HMD system is used by many projects and subprojects in a variety of configurations. The following is an overview of the current system:

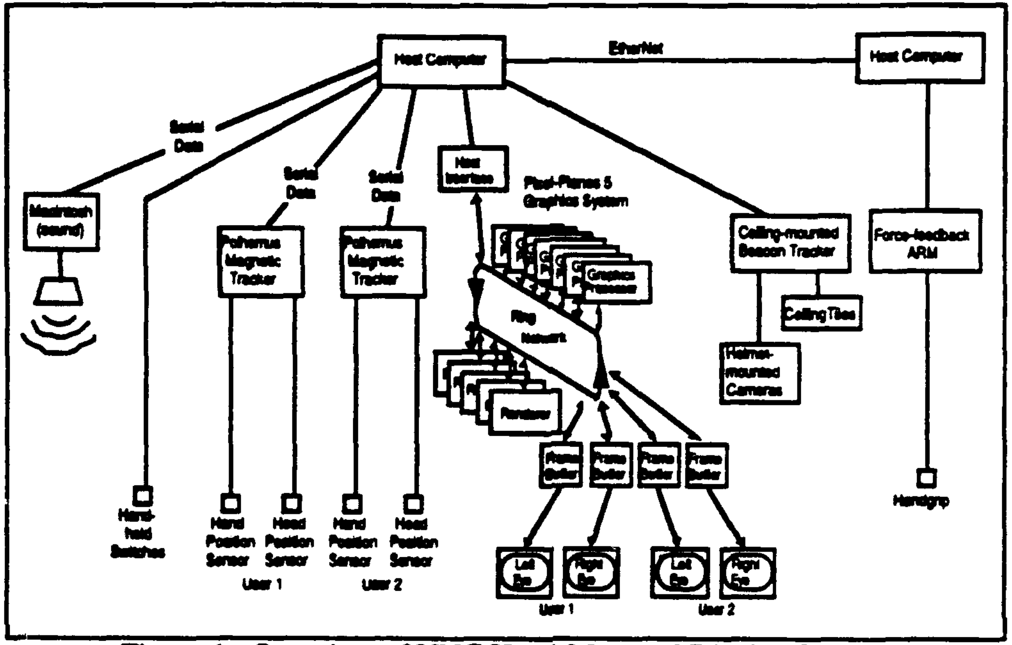


Figure 1. Overview of UNC Head-Mounted Display System

A description of each component follows:

- Host computer: A Sun 4, which runs the host side of the application and interfaces to the graphics engine.
- Graphics Engine: Pixel-Planes 5, a message-passing multicomputer capable of drawing more than two million Phong-shaded, Z-buffered triangles per second. Multiple applications can run on the system at the same time due to its ability to be split into multiple subsystems.
- Magnetic trackers: Currently 3SPACEs and Birds, but we will be beta testing new models soon, and hope to increase our system's performance with faster, newer models from Polhemus and/or Ascension.
- Ceiling tracker: Described in Section 4.3.
- Force feedback ARM: Simulates forces in the virtual world; has six degrees of freedom (3 forces and 3 torques); uses computer-controlled servo motors.

- Manual input devices: Currently hollowed-out billiard balls with Polhemus sensors mounted inside, and switches on the outside for signaling user actions. Also, a Polhemus sensor mounted on a guitar finger pick equipped with a small switch, which is well-suited for simple manipulation tasks. We should soon have a new, four-button input device for experimentation.

- Sound: Controlled by Macintosh; plays sounds under application control either in earphones attached to HMD, or from separate speakers.

In general, an application can run with almost any subset of the above equipment.

### *Software*

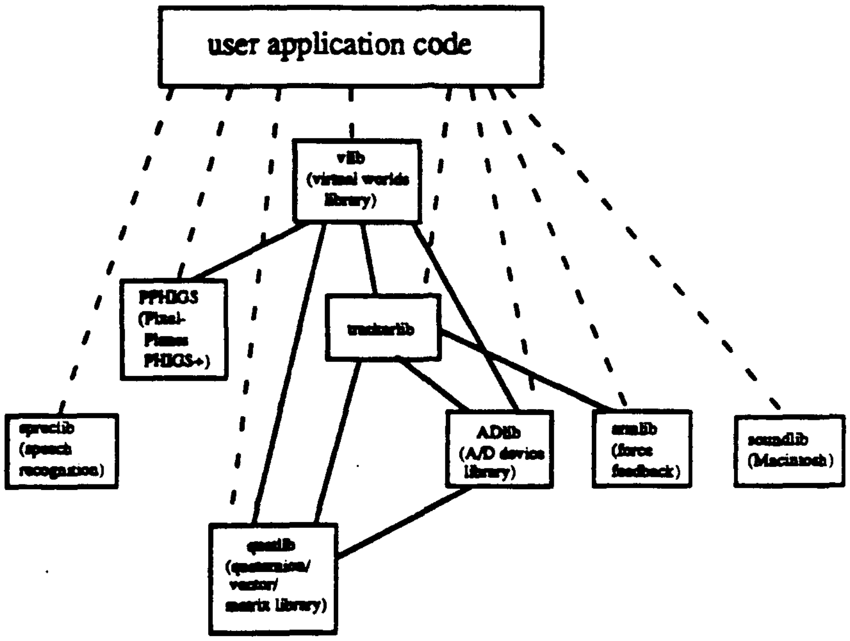

user application code
vlib (virtual worlds library)
PPHIGS (Pixel-Planes PHIGS+)
trackerlib
sprechlib (speech recognition)
ADlib (A/D device library)
armlib (force feedback)
soundlib (Macintosh)
quatlib (quaternion/ vector/ matrix library)

Figure 2. Overview of UNC Virtual-Worlds Software.
(Dashed lines indicate application calls; solid lines indicate inter-library calls.)

The virtual-worlds base software is fairly complete but will continue to expand to be more flexible and powerful and to accommodate new hardware. Most of the libraries support multiple technologies transparently, allowing application code to switch easily between different pieces of hardware by changing only an environment variable. While efforts at improving these libraries will continue, we will work to add new capabilities and develop new applications. The software goals are listed in with the other twelve-month goals for the project in Section 3.1.

## 4.2 Optics

We successfully completed the design of a 30-degree field-of-view (FOV) optical system using off-the-shelf optical elements. The system consists, for each eye, of a semi-transparent mirror folded at 45 degrees and of two off-the-shelf lenses. In the head-mounted version of this system, we will use a 240x360 color LCD display with a display area of 54 x 40 mm. The effective resolution, however, is 240x120 because of the triad pixels used for color. A ray tracing of the system is shown in Figure 4.1.

This system was assembled and tested on the optical bench using a pair of stereo slides as the display, and it is now being assembled in a helmet configuration. The main advantage in building this prototype, despite a FOV limited to 30 degrees, is to test some of our ideas in terms of helmet designs with optics integration. This prototype will be lightweight and of low cost. It is also expected to be functional within a few months, which will allow us to study more thoroughly perception issues related to see-through HMDs. We also plan to set up another version of this prototype on the optical bench in order to carry out specific perceptual studies that do not require walking around in space, such as perceived depth and size in such a "semi-virtual" world. Having a bench prototype can also allow more flexible experiments and adjustments that may not be available in the headset.

The optics for a 60 degree field-of-view HMD with a tilted catadioptric combiner have been designed, and we are preparing to begin building it in a few months. The basic design is composed of a tilted catadioptric combiner, a relay lens, and a one-inch CRT color display. An optical layout of the lens is shown in Figure 4.2. The most prominent problems or difficulties with most wide-angle HMDs are: small eye relief; small pupil sizes; vignetting at the edge of the FOV; the increase in weight due to the necessity for large optical components, as well as the loss of image quality, as the pupil size increases; the loss in transmission at the optical combiner level; and spurious reflections from unwanted sources in the environment. This system will allow for the wearing of most eye glasses, and the pupil size will be sufficient to allow rotation of the eye in a cone of 60 degrees without causing vignetting of some part of the FOV. One of the main advantages of our system design is expected to be freedom from spurious reflections.

The novelty in our approach to a wide-angle, see-through HMD is the use of a catadioptric combiner tilted at about 30 degrees from the optical axis of the eye when looking straight ahead. Such a tilt of the combiner can lead to complex optical systems for the relay lens that interface the display and the combiner. Moreover, the optical performance of a tilted system becomes asymmetric with the tilt angle, and additional tilted and decentered components are often required to achieve good image quality. We have succeeded in the design of an optical system that is composed of a single optical element combiner, five lenses for the relay lens, only two different optical glasses, and small tilts and decenters for the relay lens elements. The system is designed to have a direct transmission of 67% from the display to the eye, 30% see-through transmission, and is to be free of spurious, unwanted reflections. The ratio of reflection/transmission of the tilted combiner can be chosen according to the needs of the application. A higher direct transmission than the see-through transmission may be desirable in our applications since the brightness of the CRT is often limited. Figure 4.2 shows that the relay lens is composed of a prism-like element forcing the ray bundle to fold around the head, a triplet lens, and a field lens. The system is color corrected, through the use of two different optical glasses.

The lens shown in Figure 4.2 has been optimized for a pupil size of 10 mm and an effective eye relief of 23 mm, as specified on the figure. The performance for an unvignetted 10 mm pupil is represented by the tangential and sagittal ray fan plots for 11 representative field angles in Figure 4.3: on axis; vertical fields of ±15 degrees, ±30 degrees; horizontal fields of 15 degrees, 30 degrees; and, finally, diagonal fields of ±15 degrees and ±30 degrees. The effective pupil size at a given moment, however, is dictated by the pupil size of the eye, that is ≤3 mm in diameter for photopic vision. The ray fans are shown in Figure 4.4 for an effective pupil size of 3mm. One of the optical aberrations that increases quadratically with the FOV and linearly with the pupil size is astigmatic blur. A plot of astigmatic blur for a 3 mm pupil is shown in Figure 4.5 across a 60x60 degree FOV. The effective area used by the CRT is drawn as a circle superimposed on the field lines. This plot reveals that astigmatic blur is negligible at everything except +30 degrees diagonal FOV. A further investigation of whether or not this astigmatic blur needs to be reduced will be carried out when looking at the modulation transfer function (MTF) plots, which are widely accepted as the standard assessment of image quality.

Another optical aberration that does not affect image quality but does affect the geometry of the image is optical distortion. We already know that such an off-axis design will not be free from distortion. This is a trade-off between having an on-axis design, low distortion system, but spurious reflections among other inconvenient features, and an off-axis system with high distortion but free from unwanted on-axis features. Moreover, optical distortion can be compensated for electronically by predistorting the CRT image using a lookup table. This allows more degrees of freedom for the system which can then be optimized to reach very high resolutions. A plot of the optical distortion is shown in Figure 4.6. The plot shows a mixture of keystone and barrel distortion, as expected. We plan to correct for the distortion electronically, and some mockups of it are currently under development with members of the Pixel-Planes team. We thought one possible tradeoff might be the speed of the image update, but a closer look at the problem seems to indicate that the Pixel-Planes architecture design is such that, even with distortion correction, the update of images should not be affected. We would consider a hardware implementation of the distortion compensation if image generation speed were to become a problem.

The final analysis of image quality is shown by the diffraction limited MTF plots shown in Figure 4.7 for a pupil of 3 mm. A plot for each of the eleven representative field angles described above is given for the reference wavelength of 559 nm here since the effect of residual chromatic aberrations can be seen on the ray fan plots. We can see that the performance is exceptional for most of the field points and still acceptable at the worst astigmatic point in the field. Indeed, a high resolution of 1024x1024 corresponds to a cutoff spatial frequency of 20 cycles/mm while we show the performance up to 40 cycles/mm to demonstrate the high potential in resolution of this system. We also looked at the polychromatic MTF plots which can give some insight into the strength of residual chromatic aberrations, and the performance holds very close indeed to the monochromatic one. It is better to consider the monochromatic plots, however, in terms of judging the image quality of the designed system.

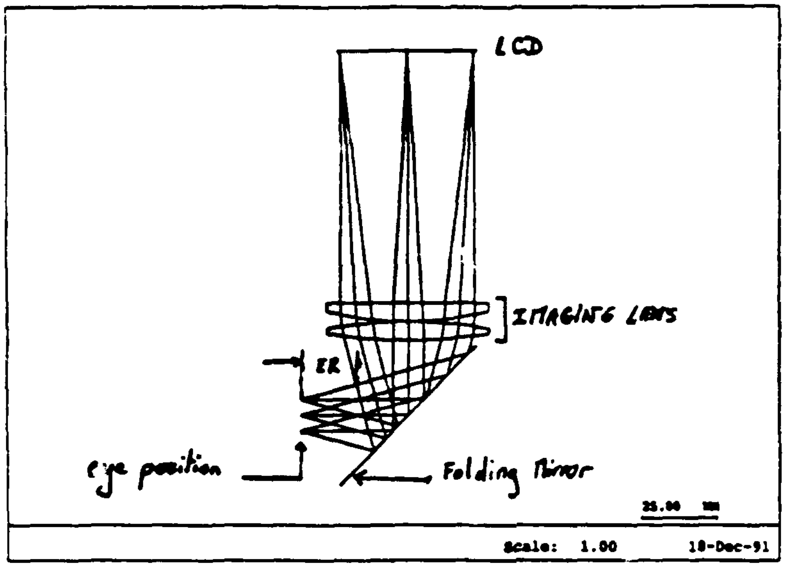


Figure 4.1. Optical Layout of the 30-degree FOV see-through HMD. The system is composed of an LCD display, an inaging lens made of two lenses, and a folding mirror. The effective eyerelief (ER), as specified in the figure, is 17mm, which should allow for the wearing of small eyeglasses.

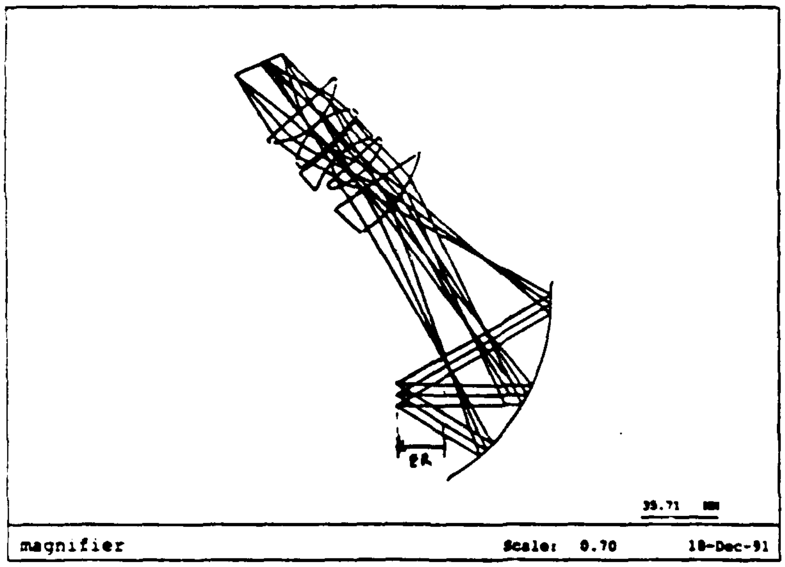


Figure 4.2. Optical Layout of the 60-degree FOV see-through HMD. The system can be thought of as composed of a tilted combiner, a relay lens made of five elements, and a one-inch CRT display. The effective eyerelief in this case is 23mm, which should allow for the wearing of most eyeglasses.

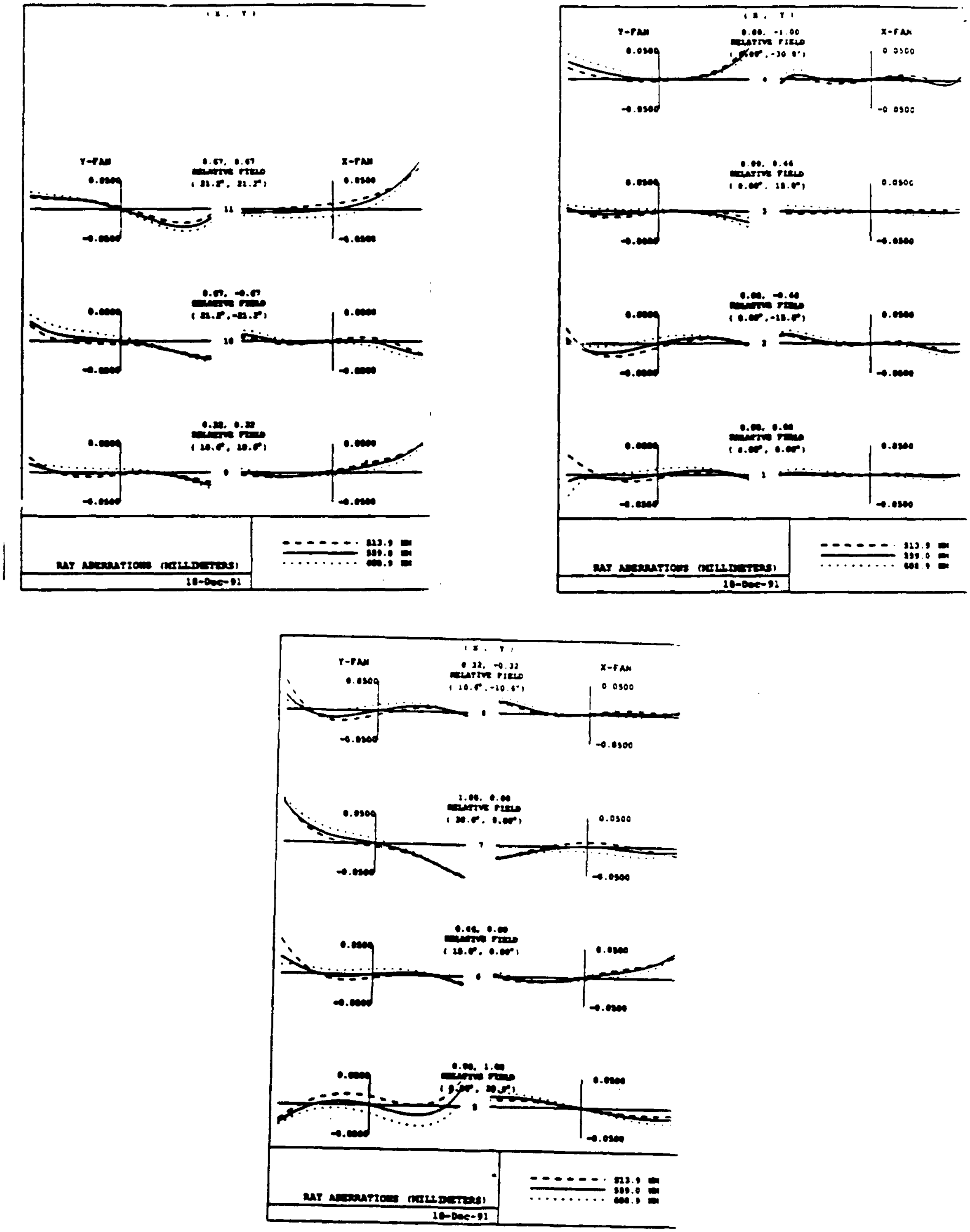


Figure 4.3. Tangential and sagittal ray fan plots for 11 field angles for a pupil size of 10mm.

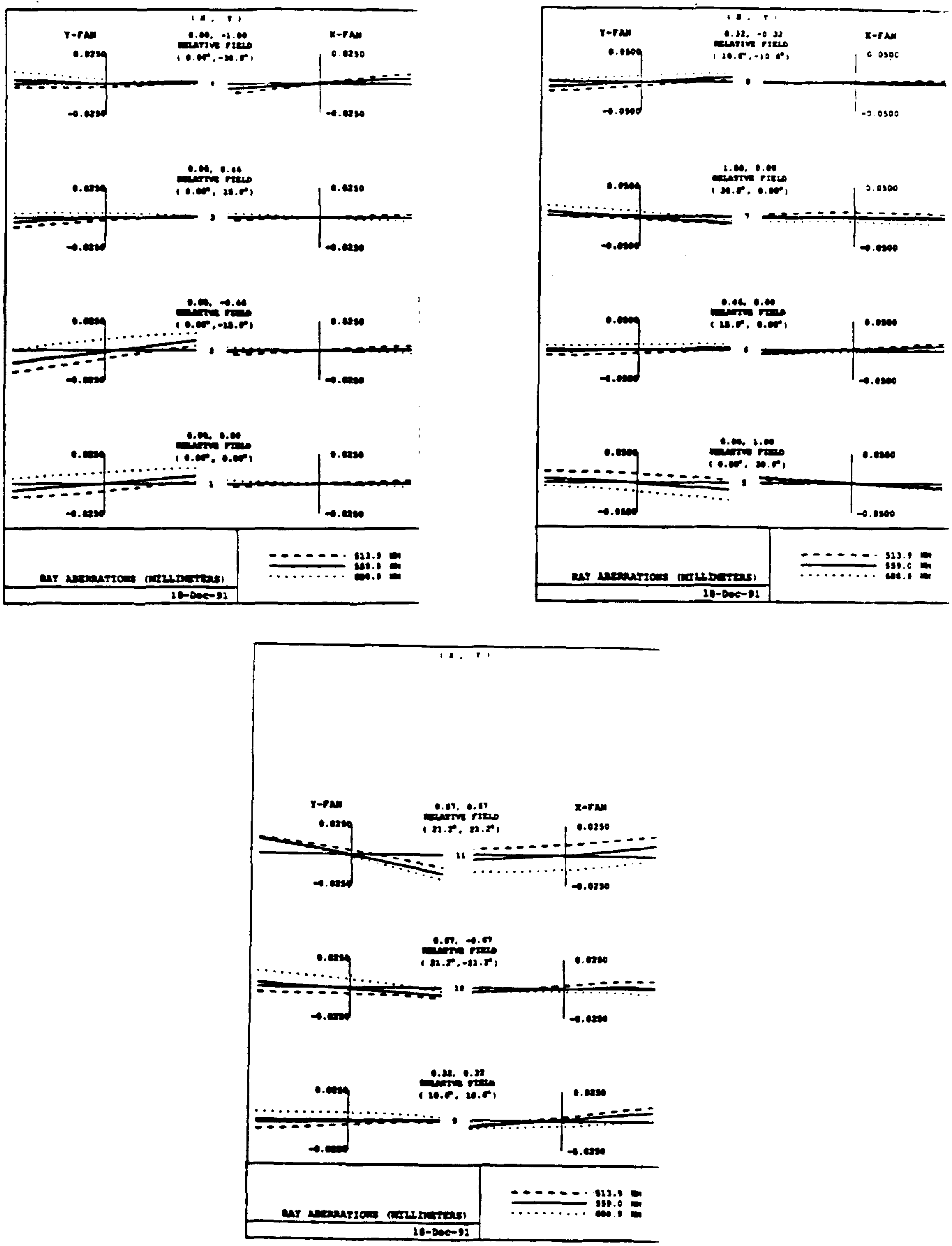


Figure 4.4. Tangential and sagittal ray fan plots for 11 field angles for a pupil size of 3mm.

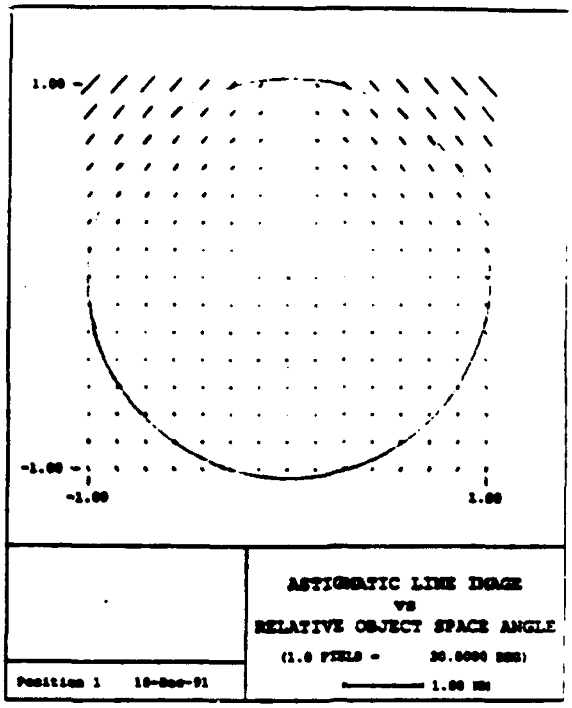


Figure 4.5. Astigmatic line image over a 60x60-degree FOV. The circle drawn on this picture represents a circular FOV of 60 degrees. This is a typical image of an optical system well compensated for astigmatism. Note the scale chosen to represent the lines.

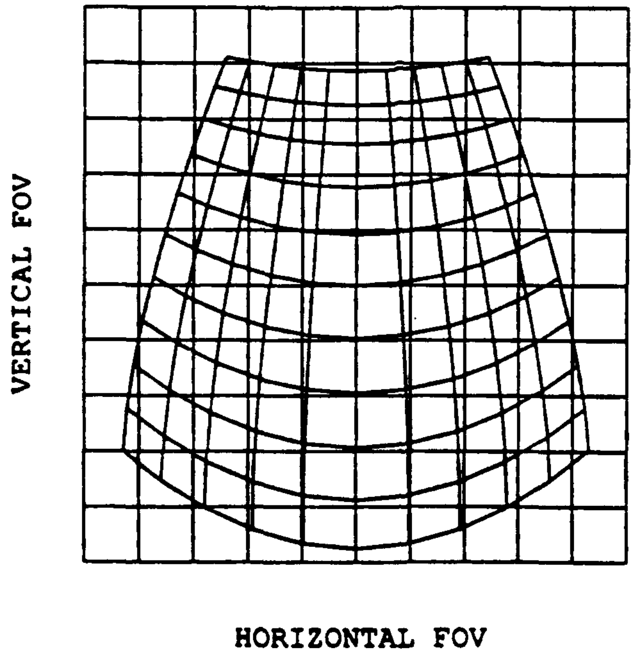
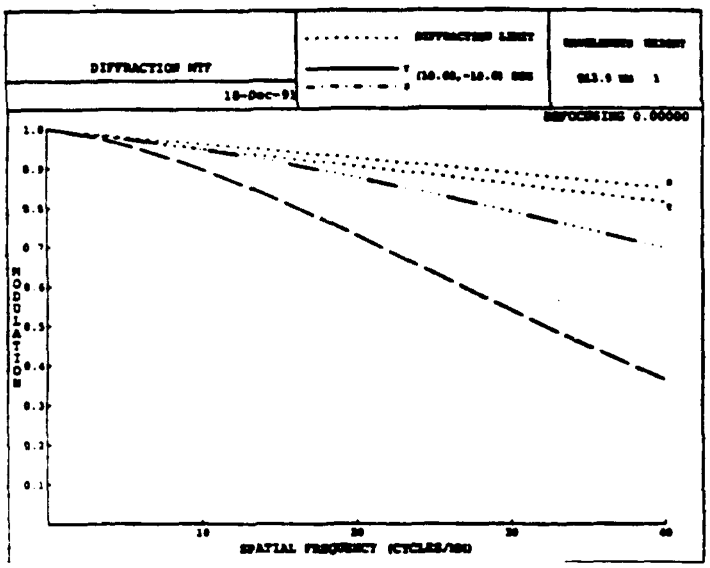
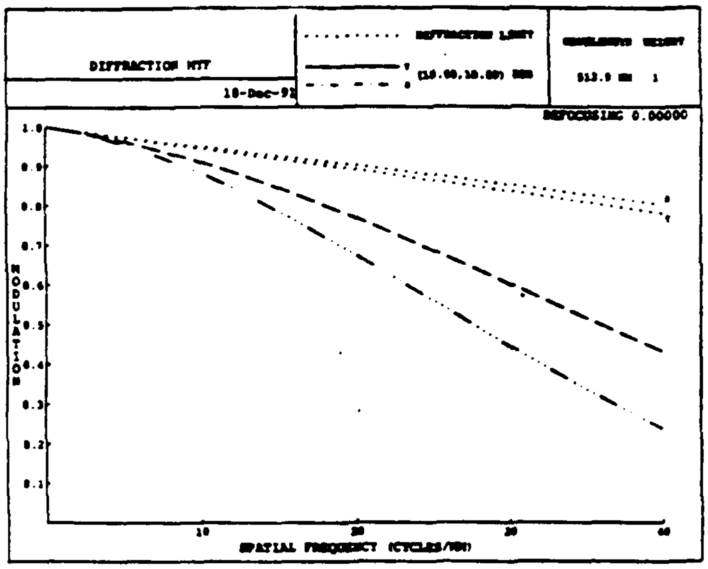
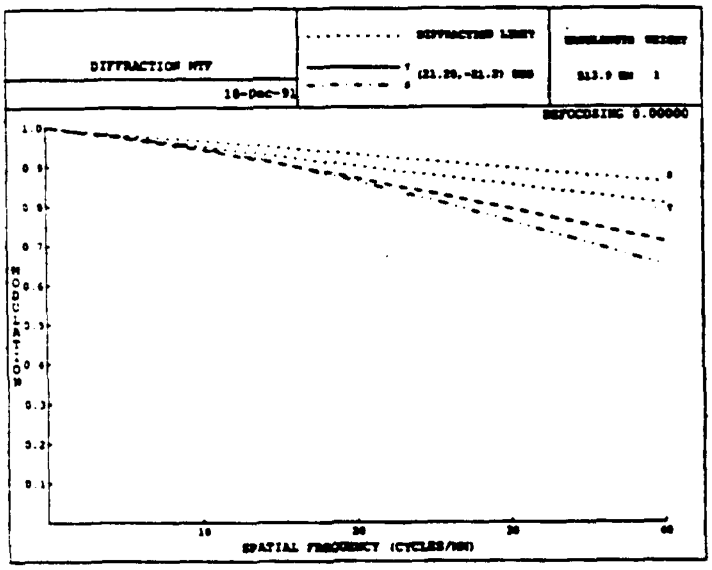
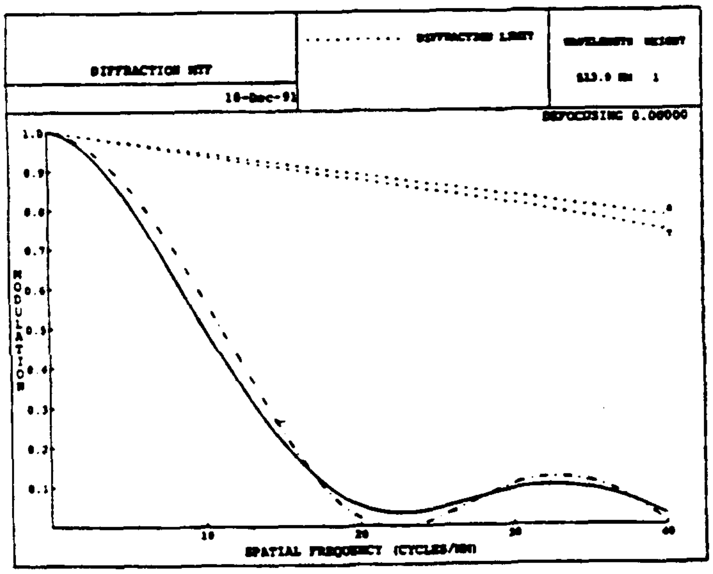


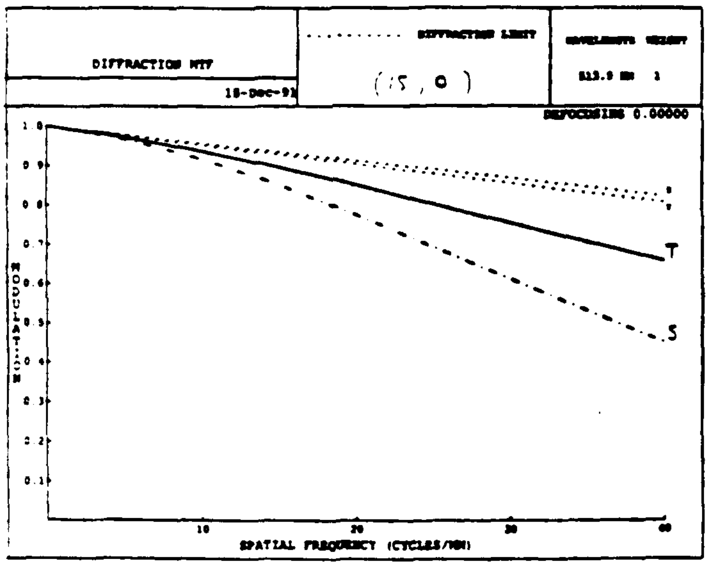
Figure 4.6. Plot of an undistorted image on the CRT when projected without any electronic distortion compensation.

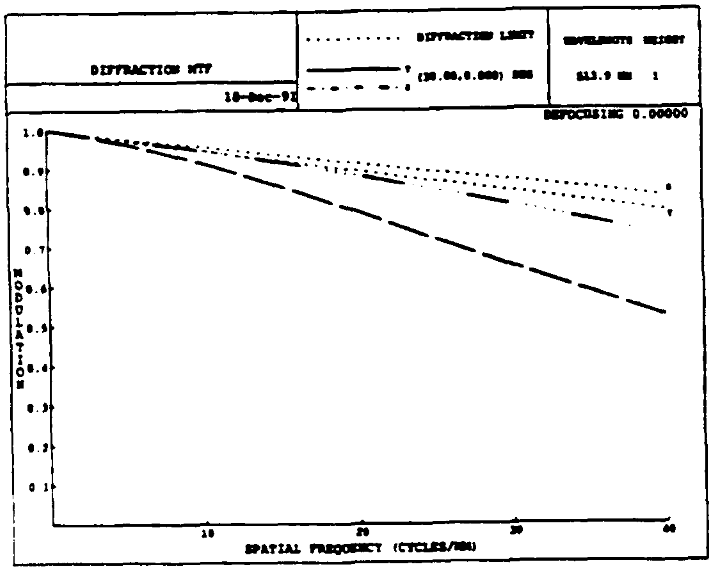

DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
T
S
(10.00,-10.00) DEG
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)


DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
T
S
(10.00,10.00) DEG
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)


DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
T
S
(21.20,-21.2) DEG
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)


DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)


DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
(15, 0)
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
T
S
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)


DIFFRACTION MTF
18-Dec-91
DIFFRACTION LIMIT
T
S
(20.00,0.000) DEG
WAVELENGTH WEIGHT
513.9 NM 1
DEFOCUSING 0.00000
MODULATION
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
SPATIAL FREQUENCY (CYCLES/MM)

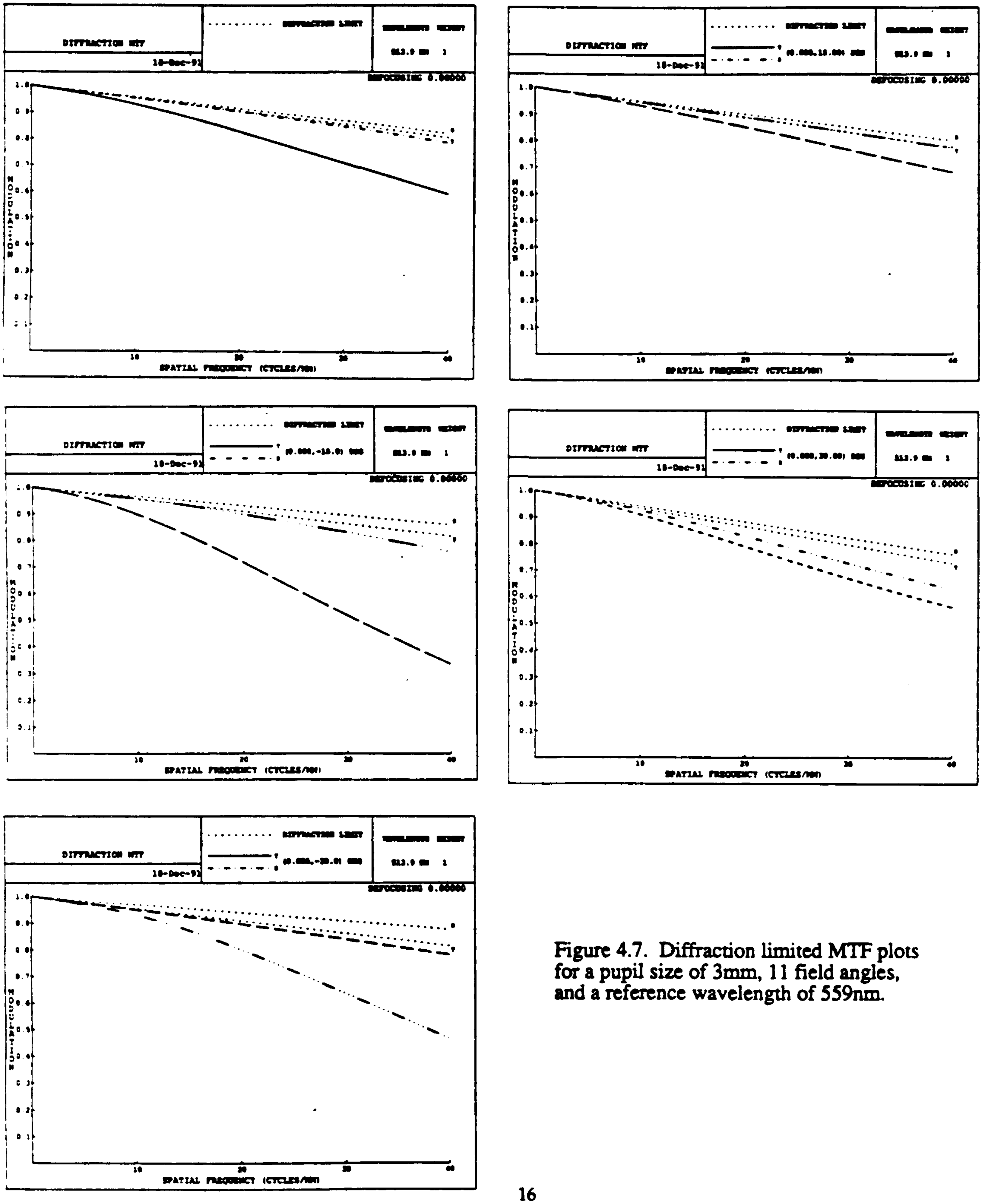


Figure 4.7. Diffraction limited MTF plots for a pupil size of 3mm, 11 field angles, and a reference wavelength of 559nm.

## 4.3 Tracking

It is well known that trackers for HMDs require high resolution, high update rate, and low latency. Range, however, has not received similar emphasis. Many virtual worlds, such as architectural walkthroughs, would benefit from more freedom of movement than is provided by existing trackers. Long-range trackers would allow greater areas to be explored naturally, on foot, reducing the need to resort to techniques such as flying or walking on treadmills. Also, such techniques of extending range work only with closed-view HMDs that completely obscure reality. With see-through HMDs, which we are attempting to build for medical applications, the user's visual connection with reality is intact and hybrid applications are possible where physical objects and computer-generated images coexist. In this situation, flying though the model is meaningless. The model is registered to the physical world and one's relationship to both must change simultaneously.

For the short-term, we have pursued an optical technology that will let us track a HMD inside a room-sized environment, which we call the ceiling tracker. The ceiling tracker gives us room-sized tracking capability today and will be a vehicle for tracker and HMD application research. But for the long term, we will pursue relative-mode trackers, such as Self-Tracker or inertial systems, that allow unlimited range tracking in unmodified environments. Since such technologies suffer from drift problems, the ceiling tracker will be used as a testbed and an aid in their development.

### *Ceiling Tracker*

The existing full-scale prototype is the second generation of an optoelectronic head-tracking concept that originated at UNC Chapel Hill. In the first incarnation, Jih-Fang Wang demonstrated the validity of the approach in a three-sensor, three-LED benchtop prototype [Wang90a]. The current system places four outward-looking image sensors on the wearer's head and locates LEDs in a 10' x 12' suspended ceiling structure of modular 2' x 2' ceiling panels. Each panel houses 32 LEDs, for a total of 960 LEDs in the ceiling. Images of LEDs are formed by lateral-effect photodiode detectors within each head-mounted sensor. The location of each LED's image on a detector, or *photocoordinate*, is used along with the known LED locations in the ceiling to compute the head's position and orientation. To enhance resolution, the field of view of each sensor is narrow. Thus, each sensor sees only a small number of LEDs at any instant. As the user moves about, the working set of visible LEDs changes, making this a *cellular* head-tracking system.

Measurements of head position and orientation are produced at a rate of 20-100 Hz with 20-60 ms of delay. Typical average values are 50-70Hz and 30ms of lag, provided that the user stands erect and keeps three or more sensors aimed at the ceiling. The system's accuracy has not been measured precisely, but the resolution is 2 mm and 0.2 degrees. It was demonstrated in the Tomorrow's Realities gallery at the ACM SIGGRAPH '91 conference, and is, to our knowledge, the first demonstrated scalable head-tracking system for HMDs.

The system is novel for two reasons. First, the sensor configuration is unique. Other optical tracking systems fix the sensors in the environment and mount the LEDs on the moving body. The outward-looking configuration is superior for it improves the system's ability to detect head rotation. The scalable work space is the system's second contribution. If a larger work space is desired, more panels can be easily added to the overhead grid.

This system extends Wang's work in several ways. First, an overhead grid of 960 LEDs was produced with well-controlled LED location tolerances, and more attention was paid to controlling other error sources as well. Second, mathematical techniques were developed that allow an arbitrary number of sensors and an arbitrary number of LEDs in the field of view of each sensor to be used in the computation of head location [Azuma91]. This resulted in an overdetermined system of equations which, when solved, was less susceptible to system error sources than the previous mathematical approach (Church's Method).

Third, the analog signals emerging from the sensors were digitally processed to reject ambient light. Finally, techniques for quickly determining the working sets of LEDs were developed.

Experience with simulations and an early 48-LED prototype revealed the problem of *beacon switching error*: as the user moved around and the working set of beacons changed, discontinuous jumps in position and orientation occurred. These are caused by errors in the sensor locations, distortions caused by the lens and photodiode detector, and errors in the positions of the beacons in the ceiling. We controlled these errors by measuring the positions of the sensors on the head unit with a coordinate measuring machine, calibrating the lens-detector assemblies, and housing the LEDs inside carefully-constructed panels and mounting those in a ceiling superstructure that was levelled with a Spectra Physics Laser-Level.

The mathematical routine used to recover the position and orientation of the user's head is a variation of a photogrammetric technique called space-resection by collinearity [Wolf83]. We have extended it to handle information from multiple distributed sensors. At least three LEDs need to be seen to compute a solution. If more are seen, this routine will generate the best solution in a least-squares sense. Using more than three beacons reduces the effects of errors from systematic error sources. The Collinearity module runs on an i860 board and takes about 6 ms to run under normal circumstances.

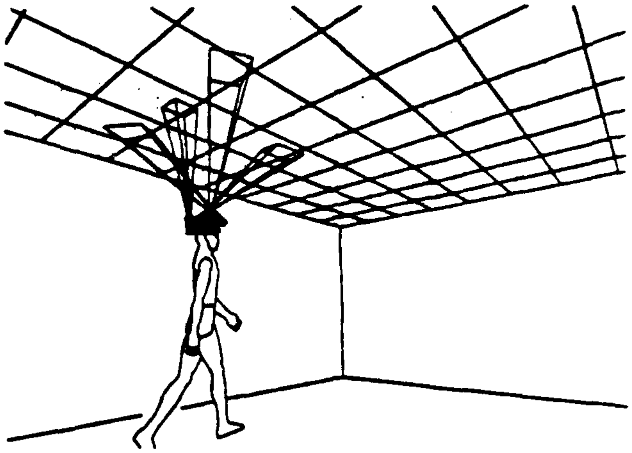
Many steps were taken in the low-level software and hardware to maintain high signal-to-noise ratios and achieve good resolution from our sensors. Ambient light was rejected by placing filters on the lenses and by taking readings with the LED off and subtracting those values from the signals with the LED on. When sensing an LED, many samples are taken and averaged to reduce random noise effects. The distance between a sensor and an LED varies widely because the user is moving around, so the current used to light an LED is adjusted dynamically to maximize signal without saturating the detectors.

At any given instant, the number of visible LEDs is small compared to the total number of LEDs in the ceiling, so discovering what this visible set is requires something faster than a brute-force scan of the entire ceiling array. Assumptions of spatial and temporal coherence let us develop more efficient heuristics: 1) Given the last known set of beacons, test those and their neighbors, if needed. 2) Given that we are computing the position and orientation of the head, use that to predict which LEDs the sensors should see if the head is actually at that location.

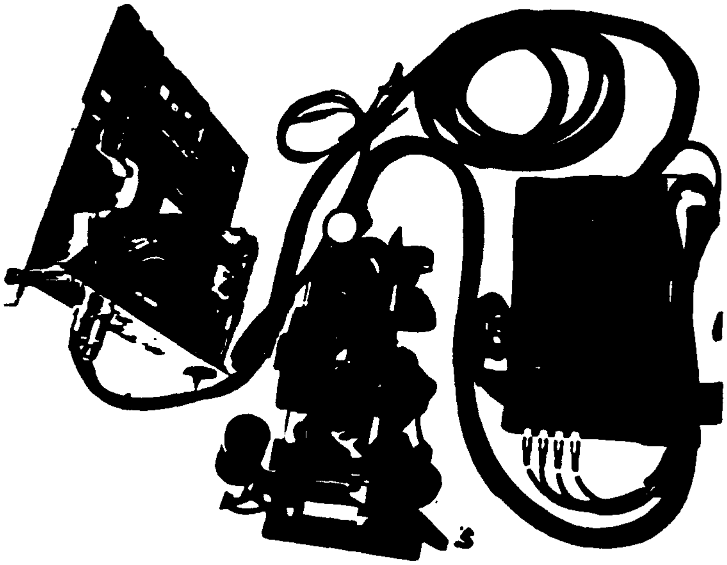
The main problem with the existing system is the combination of excessive head-born weight and limited rotation range. Rotation range depends heavily on the user's height and position under the ceiling. A typical maximum pitch range near the center of the ceiling is 45 degrees forward and 45 degrees back. Given the current focal lengths, simulations show that as many as eight fields of view are required for a respectable rotation range [Wang90b]. The existing head unit weighs about 10 lbs. (4 lbs. for the displays, 6 lbs. for the head frame and sensors). The weight of each sensor must be significantly reduced to achieve this goal. To reduce weight, we are trying to replace the current lenses (11 oz. each) with smaller, lighter lenses (2 oz. each). Wang also proposed optically multiplexing multiple fields of view onto on a single lateral-effect photodiode. However, it may be easier to design a new head unit with integral photodiodes and lenses. Given that each photodiode is about the size of a quarter, the entire surface of a head unit could be studded with sensors.

# Photos of the optical ceiling tracker

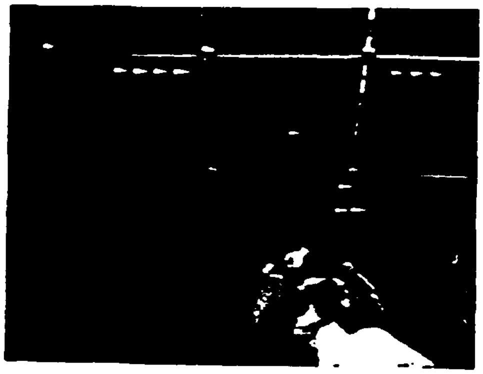
The existing system in UNC's graphics laboratory



Conceptual drawing showing fields of view of the head-mounted sensors



Head unit with four sensors and supporting electronics



Sensors viewing LEDs in the ceiling. Photo taken with a camera that is sensitive to infrared light.

### *Self-Tracker*

The self-tracker [Bishop84] is one possible inside-out tracking system which may replace, or augment, the optoelectronic system. Its goal is to eventually eliminate fixed beacons and perform tracking using features found naturally in the environment. The self-tracker could then be taken to any room, hallway, or even outdoors.

The self-tracker will use a cluster of custom made VLSI chips which will be mounted on the head-mounted display and look out on the environment in multiple directions. The cluster of chips will report the apparent shift of the image seen by each chip during successive clock cycles. A host computer will collect the information and calculate the distance the user has travelled relative to his previous position. Approximately 1000 frames per second will be required to track the fastest possible movements reported in human motion literature. This system may not work by itself because of accumulated drift. For the initial versions, a small number of beacons may be placed in the environment to allow for error correction.

Each self-tracker chip will contain a one-dimensional array of photosensors and supporting image processing circuitry. A one-dimensional array will be used so that a sufficiently large number of pixels can be placed on the chip and so that image correlation can be performed in real time. The correlation between successive frames will be accomplished by converting the image to a digital representation and then successively shifting the position of one frame and reporting the shift which corresponds to the closest match to the other frame.

The tracker chips will be mounted in pairs so that stereopsis can be used to measure the distance to the observed surface in the scene. Each tracker chip will report the observed shift in terms of the number of pixels of shift, so the distance value will be required to convert the shift from pixels to a physical measurement. One chip in each pair will transmit the its image's digital representation to the other chip. The second chip will then calculate the correlation between the two images in the same manner that it calculated the correlation between subsequent images on the same chip.

The diagram below is a block diagram of a self-tracker chip. A one-dimensional photosensor array generates an array of analog voltages which represent the incident light on the array. The edge detection block produces a digital representation of the image. Then, the image comparison block compares the current image with the image seen during the last time frame. The shift between the two images is reported, and the current frame is stored for comparison with the next frame. Also, each image is sent to its stereo partner to calculate the distance to the observed image.

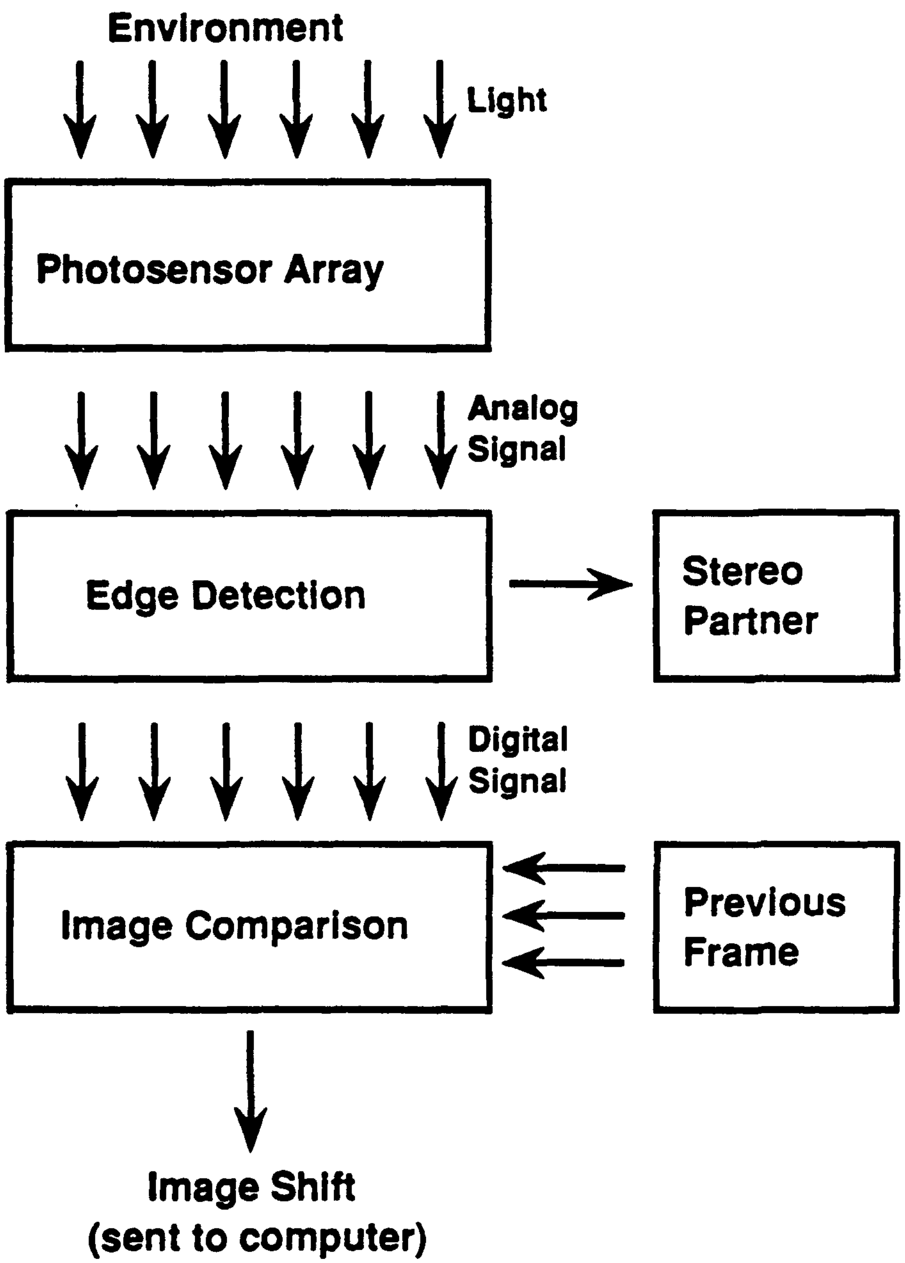


Figure 3. Architecture of the Self-Tracker custom VLSI sensor.

Over the past six months, research has progressed in both hardware and software. A set of prototype Self-Tracker chips with a complete set of image detection and processing circuitry has been fabricated through MOSIS. In addition, supporting hardware which connects the Self-Tracker chips with the host computer has been constructed. A software model of the Self-Tracker chips using images generated by Pixel-Planes 5 is under development. The software model has been added to software which uses images captured from a TV camera. The model has been used to refine image processing circuitry on the Self-Tracker chips to reduce the level of errors.

### *Inertial Technologies*

Inertial guidance systems, used to steer planes, ships, guided missiles, etc., do not require modification of the environment and have no range restrictions. They could be nearly ideal for tracking HMDs in the future, if the endemic drift problems can be controlled. Small, light accelerometers are available today; small, light laser-ring or fiber-optic gyroscopes are just on the threshold of availability to the public and are very expensive. A hybrid approach combining inertial technologies with a system that occasionally provided absolute location fixes to combat drift could be highly effective.

## 4.4 Interactive Building Walkthrough

The UNC Interactive Building Walkthrough Subproject aims at the development of systems to help architects and their clients explore a building design prior to its construction and, therefore, correcting problems on the computer instead of in concrete [Brooks86]. The scenes are shown with radiosity illumination, lights that can be switched on or off with only 100 msec delays, textured surfaces, and near-real-time display from changing viewpoints. Update rates realized by the Pixel-Planes 5 graphics scene generator are 30–40 updates/second (15–20 stereo images/second) on scenes consisting of thousands of polygons, many of which are textured.

One system uses a head-mounted display and the new UNC Ceiling Tracker to allow the viewer literally to walk freely about a 10' x 12' living room in an imaginary apartment. Both systems include the display of scene-appropriate sounds. The second system uses our UNC steerable treadmill and a head-mounted display to allow viewers virtually to walk around a fully furnished ten-room house. Different appropriate sounds are heard in each room and outdoors.

We have long held that virtual worlds projects need to move forward simultaneously in four dimensions: faster, prettier, handier, and realer. In the past year, we have made the following progress in each of these areas:

- ***Faster:***

  The Pixel-Planes 5 machine came on-line last year. Over the course of the year, we and the Pixel-Planes 5 team worked together to remove the initial hindrances at realizing its speed. We transported Pixel-Planes 5 to the SIGGRAPH '91 conference and demonstrated its ability to display 2 million Phong-shaded, on-screen triangles/second; this is about twice as fast as the state-of-the-art Silicon Graphics machine. Texturing substantially reduces this rate. The power of Pixel-Planes 5 is, however, sufficient for dynamic interaction of high-fidelity images.

- ***Prettier:***

  The major step forward this year has been the addition of anti-aliased textures. They radically improve the visual fidelity of a scene with a fixed number of polygons. We have added textures for wood, brick, ceiling tile, linoleum, bathroom tile, etc. We even use textures for sheet music, piano keys, and art hanging on the wall. Pixel-Planes 5 is very good at procedural textures; it is not especially good at bit-mapped (image) textures.

  The second major step on scene fidelity has been the addition of location-dependent (monaural) sound cues. We adapted commercial Macintosh software, purchased a 10,000-sound CD sound-cue library, and developed the location-based sound selection, loudness control, and sound–sound mixing software.

- ***Handier:***

We developed new user interfaces for:
- the ceiling tracker
- interactive turning on and off of lights in the model
- full integration of our treadmill interface with the head-mounted display and Polhemus head tracker.

- ***Realer:***

We gained a lot of experience in model building this year. The house model was refined, and a sound model was built. We also built *de novo* a model for a six-room furnished apartment. On the theoretical side, we developed a whole, new, many-level hierarchical concept of how architectural models should be organized.

In the past, only the display stage has been interactive, and that interactivity has been limited. Although the lights could be controlled interactively, the geometry was fixed. Any changes in the model had to be sent back through the model pipeline, a process that could take days or weeks. Changes in the model were typically limited to corrections of modelling errors; changes in design were prohibitively time-consuming.

Much of ou present activity is devoted to improving the system interactivity, to decreasing the time between model design-cycle iterations. We have begun a collaboration with Virtus Corporation, using their package, Virtus WalkThrough, as a modelling tool. They have provided a version of Virtus WalkThrough from which we can export a model to Pixel-Planes 5 for immediate display. One of our goals is to be able to do interactive design, with a user under the UNC Ceiling Tracker making design decisions and a modeller using Virtus WalkThrough to carry them out. The user will be able to explore a model, manipulate objects, suggest modifications, and have those modifications implemented in the model around them.

**References:**

[Azuma91] Azuma, Ronald and Mark Ward. Space-resection by collinearity: mathematics behind the optical ceiling head-tracker. University of North Carolina at Chapel Hill Department of Computer Science technical report TR 91-048, November 1991.

[Bishop84] Bishop, Gary and Henry Fuchs. "The Self-Tracker: A Smart Optical Sensor on Silicon," *Proceedings of the 1984 MIT Conference on Advanced Research in VLSI* (Dedham, MA: Artech House, January 1984), pp. 65-73.

[Brooks86] Brooks, Jr., F. P. "Walkthrough - A dynamic Graphics System for Simulating Virtual Buildings," *1986 Workshop on Interactive 3D Graphics*, S. Pizer and F. Crow, eds., University of North Carolina at Chapel Hill, October 1986, pp. 9-22.

[Rhoades92] Rhoades, John, Greg Turk, Andrew Bell, Andrei State, Ulrich Neumann, and Amitabh Varshney. "Real-Time Procedural Textures", to appear in the *ACM Computer Graphics,* Proceedings 1992 Symposium on Interactive 3D Graphics.

[Varshney91] Varshney, Amitabh. "Parallel Radiosity Techniques for Mesh-Connected SIMD Computers," University of North Carolina at Chapel Hill Department of Computer Science technical report, TR91-028.

[Wang90a] Wang, Jih-Fang, Vernon Chi, Henry Fuchs. "A real-time 6D optical tracker for head-mounted display systems," Proceedings of 1990 Symposium on Interactive 3D Graphics (Snowbird, Utah, 1990). In *Computer Graphics* 24, 2 (March 1990), pp. 205–215.

[Wang90b] Wang, Jih-Fang, Ronald Azuma, Gary Bishop, Vernon Chi, John Eyles, Henry Fuchs. "Tracking a head-mounted display in a room-sized environment with head-mounted cameras," *SPIE Proceedings*, Vol. 1290, *Helmet-Mounted Displays II* (Orlando, Florida, April 19–20, 1990), pp. 47–57.

[Wolf83] Wolf, Paul. *Elements of Photogrammetry, With Air Photo Interpretation and Remote Sensing*, 2d ed., McGraw-Hill, New York, 1983.

## 5.0 Dissemination of Research

### 5.1 Publications

Butterworth, Jeff, Andrew Davidson, Stephen Hench, and T. Marc Olano. "3DM: A Three-Dimensional Modeler Using a Head-Mounted Display." To appear in *ACM Computer Graphics—Proceedings 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

Chung, James. "A Comparison of Head-tracked and Non-head-tracked Steering Modes in the Targeting of Radiotherapy Treatment Beams." To appear in *ACM Computer Graphics—Proceedings 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

Holloway, Richard, Henry Fuchs, and Warren Robinett. "Virtual-Worlds Research at the University of North Carolina at Chapel Hill." To appear in *Proc. Computer Graphics '91*, London, November 1991.

Rhoades, John, Greg Turk, Andrew Bell, Andrei State, Ulrich Neumann, and Amitabh Varshney, "Real-Time Procedural Textures." To appear in *ACM Computer Graphics—Proceedings 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

Robinett, Warren. "Electronic Expansion of Human Perception." *Whole Earth Review*, Fall 1991, pp. 16-21.

Robinett, Warren, and Richard Holloway. "Implementation of Flying, Scaling, and Grabbing in Virtual Worlds." To appear *ACM Computer Graphics—Proceedings 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

Robinett, Warren, and Jannick Rolland. "A Computational Model for the Stereoscopic Optics of a Head-Mounted Display." To appear, *Presence*, vol. 1, no. 1, Jan. 1992 (a new refereed journal devoted to virtual reality and telepresence). Also published as UNC Dept. of Computer Science technical report TR91-009, Feb. 1991.

Varshney, Amitabh, "Parallel Radiosity Techniques for Mesh-Connected SIMD Computers," UNC Department of Computer Science technical report TR91-028.

Ward, Mark, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs. "A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems." To appear in *ACM Computer Graphics—Proceedings of 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

## 5.2 Presentations

By: Frederick P. Brooks, Jr.
Topic: Computer Graphics: More Unsolved Problems
Event: SIGGRAPH '91 panel
Place: Las Vegas, NV
Date: July 28–August 2, 1991

By: James Chung
Topic: Application of Head-Mounted Display to Radiotherapy Treatment Planning
Event: CHI'91 Doctoral Consortium
Place: New Orleans, LA
Date: April 27–May 2, 1991

By: Henry Fuchs
Topic: Scientific Visualization on Advanced Architectures
Event: SIGGRAPH '91 panel
Place: Las Vegas, NV
Date: July 28–August 2, 1991

By: Richard Holloway
Topic: Virtual-Worlds Research at UNC-CH
Event: Computer Graphics '91
Place: London, England
Date: November 1991

By: Warren Robinett
Topic: Head-Mounted Display Research at NASA and UNC-CH
Event: Banff Centre for the Arts
Place: Banff, Alberta, Canada
Date: October 30, 1991

By: Warren Robinett
Topic: Concrete Portrayal of Mathematical Abstraction Using Computer Graphics
Event: SIGGRAPH '91
Place: Las Vegas, NV
Date: August 1991

By: Warren Robinett
Topic: Expansion of Human Perception
Event: SRI Conference on Virtual Reality
Place: Menlo Park, CA
Date: June 1991

By: Warren Robinett
Topic: Virtual Reality for Education
Event: International Conference on Education and Technology
Place: Toronto, Canada
Date: May 1991

By: Jannick Rolland
Topic: Insight into Optical Properties of Wide-angle See-through Head-Mounted Displays
Event: SRI Conference on Virtual Reality
Place: Menlo Park, CA
Date: June 1991

## 5.3 Some of the Visitors who observed HMD demos at UNC-Chapel Hill

- Six from National Center for Supercomputing Applications
- Six from the Math and Science Center, Richmond, Virginia
- Daniel Goldstein, Swiss journalist
- Eugene Bylinsky, Fortune Magazine
- Architecture students from Guilford College, Greensboro, NC
- Larry Engel, WGBH-TV, Boston, Mass.
- Cindy Copolo, UNC Center for Math and Science Education
- Ruth Pachter, Tim Bunning, and Dave Zelman, Wright-Patterson Air Force Base
- Gary Diver, Jim Davis, and Bill Tollhurst, IVEX Corp.
- Gottfried Ungerboeck, Mark Rinaldi, Jim Gray, IBM
- Dr. Karl Zehnder, ETH-Zürich
- Ivan Sutherland and Robert Sproull, Sun Microsystems
- Charles Grimsdale and Phil Atkins, Division Ltd., Bristol, UK
- Capt. Don Stark and Gloria Golden, Fort Huachuca, AZ
- Vic Reis and John Toole, DARPA/ISTO
- Attendees of the DARPA Prototyping Spring Contractors Meeting (30-35 people)
- Richard Clinch, North Carolina Board of Science and Technology
- Chief Academic Officers from about 30 East Coast Universities
- Laurent Belsie, Christian Science Monitor
- Attendees of the International Meeting of the Molecular Graphics Society (around 125 people)
- Dennis Roark and Jim Rautner, SAIC
- 30 Students from Program for Future Math and Science Teachers, sponsored by NC Governor's Offi
- Randall Shumaker and Dale Long, Naval Research Lab
- Bruce Phillips, North Carolina Museum of Life and Science
- John Lang, Hewlett Packard
- Jay Schadler, Kim Spencer, and crew from ABC-TV's Prime Time Live
- John Imholz, Virginia Tech School of Architecture
- Tokiichiro Takahashi, NTT, Japan
- Phil Amburn, Air Force Institute of Technology
- Dave Neyland, DARPA
- Andy Goris, Hewlett Packard
- Victor Zadolotnyi and Michael Hovan, George Mason University
- Bill Furness and Martin Green, IST
- Steve Ellis, NASA Ames Research Center
- Takayuki Nakajima, Sharp Corp., Tokyo
- Ben Delaney, Cyberedge Journal
- Len Bass and Rick Kazman, Software Engineering Institute, Carnegie Mellon University
- Sat Narayanan and Richard Knapp, Tektronix
- John Hestenes, National Science Foundation
- Dr. Merten, President, Siemens Research Corporation
- Dr. Ming-Yee Chiu, Head of Imaging Dept., Siemens Research Corp.
- Lance Glasser and Marko Slusarczuk, DARPA
- Ten from SAS Institute, Cary, NC
- Rich Garner, Bob Liang, and Dan Ling, IBM T. J. Watson Research Center
- Capt. Brad Hooker, Fort Huachuca, AZ
- Eric Melvin and Jane Fanning, Digital Equipment Corp.
- Wayne Winstead and Jorge Montoya, Research Triangle Institute, RTP, NC
- Joe La Russa, Electro Visual Engineering, Yorktown Heights, NY
- Wim van Damme, Univ. of Utrecht, The Netherlands

## 6.0 Appendixes

### Appendix A

Paper: Butterworth, Jeff, Andrew Davidson, Stephen Hench, and T. Marc Olano. "3DM: A Three-Dimensional Modeler Using a Head-Mounted Display." To appear in *ACM Computer Graphics—Proceedings of the 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

### Appendix B

Paper: Chung, James C. "A Comparison of Head-tracked and Non-head-tracked Steering Modes in the Targeting of Radiotherapy Treatment Beams." To appear in *ACM Computer Graphics—Proceedings of the 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

### Appendix C

Paper: Robinett, Warren, and Richard Holloway. "Implementation of Flying, Scaling, and Grabbing in Virtual Worlds." To appear in *ACM Computer Graphics—Proceedings of the 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

### Appendix D

Paper: Ward, Mark, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs. "A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems." To appear in *ACM Computer Graphics—Proceedings of the 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., April 1992).

### Appendix E

Handouts from the Head-Mounted Display demonstrations in the Tomorrow's Realities Gallery at the SIGGRAPH '91 conference in Las Vegas, NV, (28 July - 2 August 1991).

- Interactive Building Walkthrough Using the Optical Tracker
- 3dm: A Two-Person Modeling System
- Flying Through Molecules
- Interactive Building Walkthrough Using A Steerable Treadmill
- Radiation Therapy Treatment Planning
- Mountain Bike
- Molecule Museum

# 3DM: A Three Dimensional Modeler Using a Head-Mounted Display

Jeff Butterworth, Andrew Davidson,
Stephen Hench and T. Marc Olano

Department of Computer Science, Sitterson Hall
University of North Carolina
Chapel Hill, NC 27599-3175

## Abstract

3dm is a three dimensional (3D) surface modeling program that draws techniques of model manipulation from both CAD and drawing programs and applies them to modeling in an intuitive way. 3dm uses a head-mounted display (HMD) to simplify the problem of 3D model manipulation and understanding. A HMD places the user in the modeling space, making three dimensional relationships more understandable. As a result, 3dm is easy to learn how to use and encourages experimentation with model shapes.

## 1 Introduction

The use of interactive 3D environments has increased the demand for complex 3D models.[9] The 3D environments that provide a sense of telepresence or "virtual reality" require a large number of models in order to give the user the illusion of being in a specific place. This demand for more models has highlighted the fact that most modeling systems are difficult to use for all but a small number of experts.[9] Through identification and removal of some of the fundamental obstacles to modeling we hope to make it accessible to more users.

Typical techniques used to select and display objects are a major hindrance to 3D modeling.[3] To place an object in 3D requires six parameters: the position (three) and the orientation (three). Most modeling systems (modelers) must settle for a 2D mouse augmented by a keyboard for this purpose. This mismatch results in difficult placement and picking of objects in modeling space. The display of models usually takes the form of a projection onto a 2D monitor. This has the effect of making spatial relationships unclear. Technological improvements to 3D model display and manipulation hardware can remove these barriers to model creation and understanding.

Current virtual reality technology provides one solution to more intuitive modeling. A HMD system gives the ability to understand complex spatial relationships of models by placing the user in the model's world. Within this type of system, a hand-held pointing device supplies users with the ability to specify 3D relationships through direct 3D manipulation. As a result, the user can build the virtual world from *within* the virtual world.

Our source of inspiration for designing a user interface for a HMD-based modeler is the current software used for 2D modeling. At one time, creating 2D models required cumbersome CAD programs. This software took a long time to learn and often did not provide real-time interaction. Now, however, 2D drawings can be manipulated by even the most casual users of personal computers. This revolution is in part the result of intuitive drawing programs like MacDraw. One of the keys to MacDraw's success is its inherent simplicity. Most work done with it requires no reading or use of the keyboard. Rather, it provides a palette of tools which is always available next to the model. To change modes, the user simply selects the tool from the palette using the mouse. The process of 3D modeling can become more accessible if some of the lessons learned from this .evaluation of 2D modeling can be applied to 3D modeling systems.

This paper presents a HMD-based system called **3dm** which simplifies the task of 3D modeling by implementing the concepts introduced above. Basic techniques for working within 3dm's virtual world are described to show how users access the various features. The implementation of 3dm is described through a presentation of its most useful commands. Finally, the results of actually using 3dm are presented with an emphasis on new techniques that can be applied within other virtual worlds.

## 2 Prior Work

A large body of work has been done on 3D modeling. Although 3D *input* devices have been used to enhance modelers, very little modeling has been done with a HMD. Some examples of modeling with six degree-of-freedom input devices are [1] and [8], but both of those used traditional 2D displays. Previous uses of HMD systems have concentrated more on exploration of virtual worlds rather than creating or modifying them. Some examples of this work with HMD's can be found in [5].

Modeling using a HMD system has been explored by Clark.[4] Users of Clark's system created parametric surfaces by manipulating control points on a wire-frame grid. This system highlighted the utility of using a HMD for improved understanding and interaction with models. Like Clark's system, 3dm relies on a HMD to help simplify modeling, but 3dm's intuitive user interface design also makes it easy to learn and use.

# 3 Implementation

3dm was developed using a VPL eyephone as the display device and Polhemus trackers to track the head and hand. A 6D 2-button mouse, developed at UNC-CH, was the input device. The images were rendered using the Pixel-Planes 4 and Pixel-Planes 5 high-performance graphics engines developed at UNC-CH.[6][7] Currently, all models created with 3dm are made up of hierarchical groups of triangles.

## 3.1 User Interface

In addition to the model, the virtual world of 3dm contains the components of the user interface. The most important of these are the toolbox and the cursor. The cursor follows the position of the hand-held mouse, giving the user a sense of hand position in the modeling space. The toolbox is the means by which most actions are performed.

Some of the user interface components are simply helpful markers that can be turned off, unlike the toolbox and the cursor, which are always visible. The user stands on a "magic carpet" which marks the boundaries of where the tracking system operates. Remaining within tracker range is important because the virtual world will begin to tilt as the user moves farther out of range. Below the magic carpet lies a checkered ground plane, above which the model is usually created. Additional reference objects, such as coordinate axes, can be turned on by the user.

Figure 1: The toolbox as seen by the user.

The toolbox initially appears suspended in space near the user's waist, but it can be moved to a more convenient location. The toolbox remains attached to the user as he or she moves around the modeling space, or it can be disconnected and left anywhere above the magic carpet. The toolbox is organized into cells containing 3D icons. Each icon represents either a tool, a command, or a toggle. Many of these icons can optionally appear in pulldown menus at the top of the toolbox in order to reduce clutter.

Icons perform actions when they are selected with the cursor. Tools change the current mode of operation as reflected in the shape of the cursor. For instance, when the user reaches into the toolbox and selects the flying tool, the cursor takes the form of an airplane. Selecting a command performs a single task without changing the current mode of operation. Toggles change some global aspect of 3dm. An example is the snap to grid toggle, which restricts cursor movement to a 3D grid when it is on.

Exploring the model provides understanding of its 3D shape, so 3dm supports multiple methods of navigating in the modeling space. The HMD system used for 3dm allows the user to walk through the model space a few paces in any direction. Walking simply does not provide the range of movement needed for most models, so 3dm supports "flying," a commonly used method of traveling through virtual worlds.[2] Flying consists of translating the user through model space in the direction that the cursor is pointing. Flying moves the magic carpet, which carries the user and the toolbox along. A method of navigation that is the complement of flying is "grabbing" the world. Grabbing the world allows the user to attach the modeling space to the cursor and then drag and rotate it. Grabbing can be used to bring a feature of the world to the user rather than forcing the user to walk or fly to the feature.

Models often require manipulation at vastly different scales. To facilitate this type of work, the user can be scaled using a process called *growing* and *shrinking*. This scaling does not affect the model: it changes the user's relative size with respect to the model. The user could shrink down to bird size in order to add eyelashes to a model of an elephant and then grow to the size of a house to alter the same model's legs. Since the user can become disoriented by all of these methods of movement, there is a command that immediately returns the user to the initial viewpoint in the middle of the modeling space.

The user receives continuous feedback in a variety of ways. The HMD system provides all visual input to the user, so the display must be updated between 15 and 30 times per second. Even during file loading and other slow operations, the screen is updated and the head is tracked. Rubber banding is implemented in many situations: when defining a new triangle, scaling or moving an object, and extruding. Predictive highlighting shows the user what *would* be selected if a mouse button were pressed. This highlighting is used in the toolbox, and even more importantly, when marking vertices. Whenever the cursor is near a model, the nearest vertex is highlighted, giving the user an indication of which vertex would be operated on before actually attempting the operation.

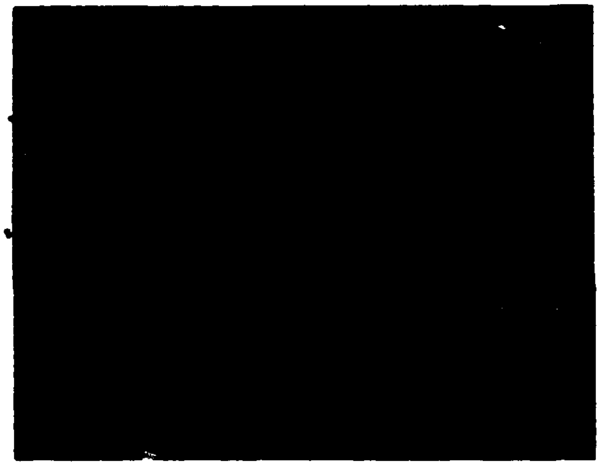


Figure 2: A triangle being added to a model. Demonstrates rubber banding and snapping to nearby vertices.

## 3.2 Tools and Commands

Although many tools are available in the 3dm toolbox, it is more useful to understand the general classes of tools supplied to the user than to enumerate all of the specific tools. Most of these tools were chosen because of their proven utility in pre-existing modelers.

### 3.2.1 Surface Creation

Surface creation is the central purpose of most 3D modeling, so 3dm provides more than one method for creating surfaces. A triangle creation tool exists for generating both single triangles and triangle strips. The corners of these triangles are specified by pointing and clicking the mouse, so the triangles are created in their desired locations rather than appearing in a "building" area and then being moved into the model space. Pre-existing vertices may be used during triangle creation to allow triangles to share corners or entire edges, making seamless connections easy.

The extrusion tool supplies a more powerful and more specialized method of triangle creation. This tool allows the user to either draw a poly-line or select one from edges already in the model and stretch it out into an extruded surface. The extrusion is performed by dragging the leading edge of the surface with the mouse. Because the mouse can be twisted and translated arbitrarily during the extrusion, it becomes easy to create complex surfaces with this tool. In addition, the leading edge of this new surface can be scaled and then extruded again as many times as necessary. This form of extrusion can rapidly create such objects as walls, legs, tree trunks, and leaves.

The last surface creation tools facilitate creation of standard surface shapes. Currently box, sphere, and cylinder tools exist. They each allow the user to interactively stretch out an arbitrarily proportioned wireframe representation of a standard shape. When the wireframe representation has the desired proportions, it is turned into a triangulated surface.

### 3.2.2 Editing

Since surfaces are rarely in exactly the desired shape upon creation, it is important that surface editing be an easy operation. The most commonly used editing tool is the mark/move tool. This tool provides a method of grasping and moving arbitrary portions of the model. Not only can entire objects be grabbed and moved with the mouse, but selected groups of vertices can be moved in order to distort part of an object. Scaling can also be performed on either entire objects or groups of vertices. During both movement and scaling, the user sees the model changing in real time. This interaction decreases the number of edits needed to make a desired change. The marking aspects of this tool are used to mark arbitrary portions of the model for operations with other tools.

Familiar editing operations from drawing programs are a group of 3dm commands that facilitate rapid experimental changes. An arbitrary number of triangles or entire objects can be cut, copied, pasted and deleted. These commands provide easy reuse of existing objects.

An undo/redo stack is provided for reversing any number of operations from any tool or command. As operations are performed, the changes they cause to the model are stored in the undo/redo stack. The undo command can then be used to pop changes off of this stack to undo as many operations as necessary. These undo operations can themselves be undone with the redo command. The undo/redo commands encourage experimental changes to the model because no operation can cause permanent damage.

### 3.2.3 Hierarchy

The hierarchical features of 3dm provide methods for organizing complex models. "Grouping" can be used to associate triangles and possibly other groups to more easily manipulate them as a whole. These groups can be *instanced*. An instance is similar to a copy of a group that can be arbitrarily translated, rotated, and scaled. However, the difference between an instance and a copy is that the instances of a group are all linked to the same basic shape. If this shape is changed, then the change is reflected in all instances at once. An example where instancing would be useful is in a model of a large building. Suppose that hundreds of chairs were in this building. If one model of a chair were instanced many times to make these chairs, then a change to a single chair would be reflected in hundreds of places throughout the building.

Groups can be organized into a hierarchy represented by a directed acyclic graph. This type of hierarchy is particularly well-suited to modeling articulated figures. The ability to instance groups and impose a hierarchy on them helps to organize models.

## 4 Results

Actual modeling sessions have shown that 3dm is efficient for rapidly prototyping models. Organic shapes, like rocks and trees, have proven to be particularly good subjects for 3dm. These shapes are easily created in 3dm because it provides a good sense for spatial relationships. Users of 3dm have commented that they feel a sense of control, because they can reach out and grab any part of the model with ease. The ability to make these quick modifications encourages the user to experiment with shapes until they are satisfactory. However, 3dm has shown weakness in the area of constraints and models that traditional CAD and drawing programs create well. For instance, 3dm has no way of keeping two polygons parallel, causing some models to appear irregular.

The extrusion tool is an example of a traditional modeling tool that has become even more powerful because of its use in a HMD framework. In most modeling systems, extrusion is performed by moving one or two spatial parameters at a time. 3dm users often alter many parameters at once during an extrusion by twisting *and* translating the new surface. Extrusion in 3dm often consists of many short extrusions. In between these short operations the leading edge of the extruded surface is often scaled and twisted. The result is that complex surfaces can be rapidly created with an easy to use tool.

Some initial solutions to 3dm's lack of constraints have been to add toggles in the toolbox for a snap-to grid and a snap-to plane. The snap-to grid constrains the position of the cursor to the nodes of a regular 3D grid. The resolution of the snap-to grid is dynamically modified to be appropriate to the user's current "grown" or "shrunk" size. The snap-to plane gives the ability to constrain cursor movement to 2 dimensions. The snap-to constraints help in making regular objects, such as mechanical parts.

## 5 Conclusion

3dm draws techniques of model manipulation from both CAD and drawing programs and applies them to modeling in an intuitive way. A HMD modeling system uses these tools to simplify the problem of 3D model manipulation and understanding.

3dm is a step toward making 3D modeling accessible to unsophisticated users. It supports users' natural forms of interaction with objects to give them better understanding of the shapes of their models. Even a novice user can understand how to manipulate a model by reaching out and grasping it. Users are encouraged to experiment with model shape because 3dm facilitates making rapid changes. The effects of a change to a modal can be clearly understood because the user can explore the model using a variety of intuitive navigation techniques.

Advanced users are also empowered by 3dm. Many of the tools borrowed from existing modeling systems become more powerful when used with a HMD. One source of increased utility is the fact that complex operations can involve simultaneous modification of many spatial parameters. Examples of tools that take advantage of this are object placement and extrusion, which both allow combinations of rotation and translation in a single step. By concentrating more functionality into each operation, fewer operations are needed to perform a task and models can be created faster.

## 6 Acknowledgements

We would like to thank Rich Holloway and Erik Erikson for VLIB and all of their software and hardware support. We would also like to thank Henry Fuchs, Fred Brooks, and the whole Pixel Planes team for supplying us with the graphical environment needed to implement 3dm.

## References

[1] Badler, Norman I., Kamran H. Manoochehri, David Baraff. Multi-Dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints. Proceedings of the Workshop on Interactive 3D Graphics, October 1986. Sponsored by ACM SIGGRAPH.

[2] Blanchard, Chuck, Scott Burgess, Young Harvill, Jaron Lanier, Ann Lasko, Mark Oberman, Michael Teitel. Reality Built For Two: A Virtual Reality Tool. *Computer Graphics*, 22(2):35-36, March 1990. Symposium on Interactive 3D Graphics.

[3] Brooks, Frederick P. Jr. Grasping Reality Through Illusion: Interactive Graphics Serving Science. *Keynote address at the Fifth Conference on Computers and Human Interaction*. Published in CHI '88 Proceedings, May 1988, 1-11.

[4] Clark, James H. Designing Surfaces in 3-D. *Communications of the ACM*, 19(8):454-460, August 1976.

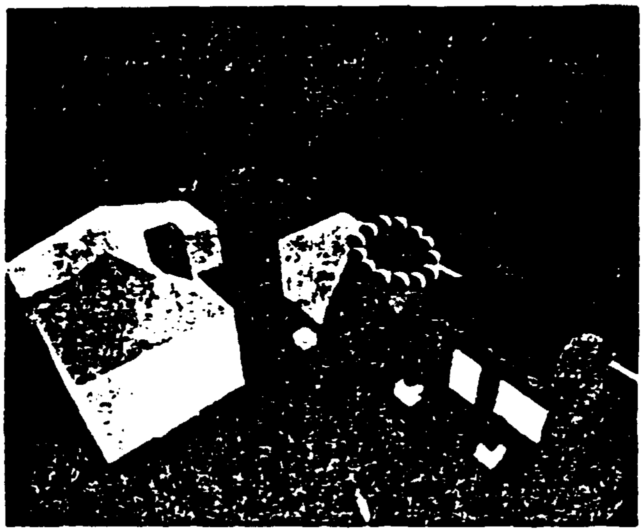
[5] Fisher, S. S., M. McGreevy, J. Humphries, W. Robinett. Virtual Environment Display System. Proceedings of the Workshop on Interactive 3D Graphics, October 1986. Sponsored by ACM SIGGRAPH.

[6] Fuchs, Henry, Jack Goldfeather, Jeff P. Hultquist, Susan Spach, John D. Austin, Frederick P. Brooks, Jr., John G. Eyles, John Poulton. Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes. *Computer Graphics*, 19(3):111-120, February 1985. SIGGRAPH '85 Conference Proceedings.

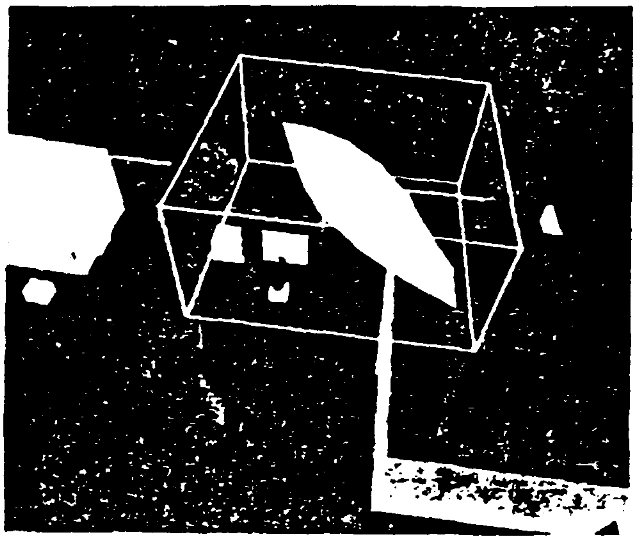
[7] Fuchs, Henry, John Poulton, John Eyles, Trey Greer, Jack Goldfeather, David Ellsworth, Steve Molnar, Greg Turk, Brice Tebbs, Laura Israel. Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories. *Computer Graphics*, 23(3): 79-88, July 1989. SIGGRAPH '89 Conference Proceedings.

[8] Sachs, Emanuel, Andrew Roberts, David Stoops. 3-Draw: A Tool for Designing 3D Shapes. *IEEE Computer Graphics & Applications*, November 1991, 18-26.

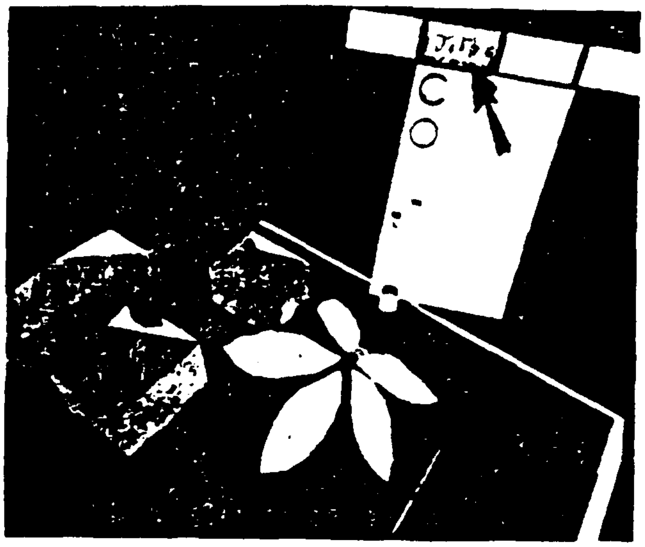
[9] Sproull, Robert F. Parts of the Frontier are Hard to Move. *Computer Graphics*, 22(2):9, March 1990. Symposium on Interactive 3D Graphics.



**Color Plate 1**
A tree trunk being extruded.
The user is roughly five times
taller than the houses.



**Color Plate 2**
A palm branch is being marked for
copying using a rubber banding box.



**Color Plate 3**
The branch has been copied four times.
Part of a toolbox menu is visible.
The red ring is the "magic carpet",
showing the tracker range.

**Color Plate 4**
The user is now normal size.
The airplane is the cursor, which indicates
that the "flying" tool is being used.

# A Comparison of Head-tracked and Non-head-tracked Steering Modes in the Targeting of Radiotherapy Treatment Beams

James C. Chung
Department of Computer Science
University of North Carolina at Chapel Hill*

## ABSTRACT

A controlled experiment was conducted to compare head-tracked and non-head-tracked steering modes in the performance of an abstract beam-targeting task. Collected data revealed a wide variety of mode preferences among the subjects. Subject performance, as measured by final score, task completion time and subject confidence, differed very little between the head-tracked steering modes taken as a group and the collective non-head-tracked modes. Some significant differences were observed between individual steering modes, both within and between the head-tracked and non-head-tracked groups.

## INTRODUCTION

Current research at the University of North Carolina at Chapel Hill is investigating the possible benefits to be gained by applying head-mounted display (HMD) technology to radiotherapy treatment planning (RTP). Use of a head-mounted display for targeting of treatment beams suggests several possible steering modes for exploring the virtual world of the patient's anatomy. To determine which steering mode is best suited to our application, a user study was conducted to investigate the relative merits of the different steering modes.

Seven steering modes were used in an abstract beam-targeting task. Four modes used head-tracking information, while the other three modes did not. It was anticipated that head-tracking would provide an advantage in beam targeting through more natural steering and navigation that makes use of proprioceptive and vestibular information, which are absent in non-head-tracked methods.

Related work in movement through a virtual world is described in [1, 2, 3], but these studies do not deal with HMD's and head-tracking.

## BEAM TARGETING

The key to successful beam targeting in radiation therapy treatment planning is to orient and shape the beams so that the entire tumor is covered by each beam while as little of the healthy surrounding tissue as possible is hit by the beams. Given the complex spatial arrangement of a patient's anatomy (tumors may be draped around healthy organs or have tendrils snaking out into the healthy tissue), this is usually not an easy task.

To evaluate the different steering modes, subjects were presented with an abstract anatomy model, consisting of a multi-colored spherical target (tumor analog) embedded in a collection of uniquely colored monochromatic spherical dodges (organ analogs). (See Color Plate 1.) The subjects used each of the seven steering modes to manipulate the direction in which a conical virtual beam passed through the model. The beam was defined such that its source (cone vertex) was always a fixed distance from the target, its central ray (cone axis) passed throught the center of the target, and its divergence was just large enough to encompass the target. The subject was instructed to find the beam direction that afforded the smallest volume of intersection between the conical beam and the dodges, a task analogous to a radiotherapist trying to avoid radiosensitive organs with a treatment beam.

## STEERING MODES

The term "steering mode" refers to the method used to change one's position or orientation in the virtual world. This is distinguished from navigation, which refers to understanding one's current position and orientation relative to other objects in the virtual world.

### Head-Tracked

These modes are linked to movement of the subject's head and enable the subject to make use of vestibular (inner ear balance) and proprioceptive (muscles, tendons, joints) senses for navigation.

*Walkaround (WLK).* In Walkaround mode, the subject physically walks about in the virtual world containing the target/dodges model. The direction of the beam is defined by the vector from the subject's eyes to the center of the target. To better examine the model and target the beam from above and below, the subject is given the ability to vertically translate the model using a 6-D mouse. No other manipulation of the model is possible.

*Walkaround/Rotation (WKR).* This is the same as Walkaround mode, with the exception that the subject is able to also rotate the model about any axis in 3-space through its center by grabbing with the 6-D mouse. (See 6-D Mouse section below.)

*Orbital (ORB).* In Orbital mode the subject is constrained to always be looking at the center of the model from the beam source. Beam direction coincides with gaze direction. Unlike the Walkaround modes, Orbital mode uses only head orientation and ignores head position. As the subject's head turns, the model is observed to translate about the subject's head at a constant distance. (Hence the name Orbital.) Because the model undergoes no rotation, it can be viewed from any direction with a turn of the subject's head.

*Immersion (IMM).* In Immersion mode the subject views the model looking outward from the center of the target. Like Orbital mode, Immersion mode makes use of head orientation only and ignores

*CB#3175, Sitterson Hall, Chapel Hill, NC, 27599-3175.
chung@cs.unc.edu (919) 962-1889

head position. When the subject's head turns, the subject's view sweeps across portions of the model from its fixed, central vantage point. The beam direction is defined by the subject's gaze direction, and the task of finding the best beam orientation becomes one of looking for the portion of the model with the biggest opening. Since the beam passes completely through the model, the subject is given the ability to reverse his gaze direction by holding down a button on the 6-D mouse, and can thereby examine the complete prospective beam path through the model.

### Non-Head-Tracked

Although these modes do not make use of head-tracking information, the subjects still viewed the model through the HMD so that image quality was equalized over the seven modes. These three modes all place the subject's eye at the beam source, looking in the direction of the beam toward the target, and support exploration of prospective beam orientations by rotating the model in three-space.

*Joystick (JOY).* In Joystick mode the model is rotated with a velocity-control joystick. In addition to the left-right/forward-backward movement of the joystick, the cap of the joystick turns clockwise and counterclockwise to provide all three degrees of rotational freedom.

*Spaceball (SPC).* In this mode the model is rotated with a Spaceball[*], an isometric, force-sensitive device that provides six degrees of translational and rotational freedom. This mode, however, uses only the three rotational degrees of freedom as a velocity control for rotation of the model in three-space.

*6-D Mouse (SDM).* In 6-D Mouse mode the orientation of the model is controlled with a custom-built, six degree-of-freedom mouse (tracker sensor embedded in a pool ball with two buttons). When either mouse button is held down, the rotational component of the mouse movement is directly linked to model rotation, and the subject sees the model rotate in the same manner as his hand.

## BEAM'S-EYE VIEW

An important feature of a steering mode that may affect a subject's performance is whether or not it provides a "beam's-eye view." Beam's-eye view is the view seen by an eye coincident with the beam vertex and whose gaze vector coincides with the beam's central axis. With a beam's-eye view it is very easy to determine which dodges intersect the beam, for since the beam is defined to diverge just enough to exactly enclose the target, the silhouettes of those dodges will overlap with the silhouette of the target. In those modes that do not provide a beam's-eye view, it is more difficult for the subject to judge which dodges are hit by the beam.

Walkaround and Walkaround/Rotate modes do not provide beam's-eye views, because the subject's head cannot be physically constrained to align with the beam source. Immersion mode also does not provide a beam's-eye view, since the subject's eyepoint is constrained to stay at the target's center. The other four modes do provide beam's-eye views.

## EXPERIMENTAL METHOD

The experiment was a one-factor within-subject investigation, with steering mode as the independent variable. Dependent variables measured were final score (volume of intersection between beam and dodges), task completion time, confidence in the final beam configuration, and rank orderings of the seven modes by ease of use and by preference.

Fourteen subjects were recruited from graduate students and staff members of the Departments of Computer Science, Radiation Oncology, and Radiology at UNC. Each subject underwent 7 sessions, each of which used a different steering mode. The order of the steering modes used by each subject was varied according to a 7x7 latin square. Each session consisted of 3 practice trials followed by 3 test trials. Each trial used a unique target/dodge model.

In each trial the subject explored prospective beam orientations until the best one was found, at which time the subject stopped the trial. There was no time limit, nor any emphasis on task completion time—the subject was instructed to take as long as necessary to find the best beam path. A virtual marker (arrow pointing through the model) was provided to the subject to use as a reference. At any time the subject could issue a "mark" command, which aligned the marker with the current beam direction, and the marker would remain fixed in the model until a subsequent command was issued. The score and task completion time for the trial were recorded, as well as the subject's rating on a scale of 1 (no confidence)-10 (total confidence) of how confident he or she was that the best beam orientation had been found.

After all seven sessions were completed, the subject ranked the seven steering modes according to two criteria, ease-of-use of the steering mode and preference for performing the beam targeting task.

Equipment used included a Polhemus 3Space[†] tracker on an EyePhone[‡] Model 2 head-mounted display, displaying images generated by UNC's Pixel-Planes 4 graphics processor.

## RESULTS

Figure 1 presents histograms showing for each steering mode, the number of times it was ranked 1st, 2nd, ... 7th by ease-of-use and by preference. The plot for Walkaround Mode shows that most subjects found it to be one of the more difficult steering modes to use. The other three head-tracking modes have somewhat flat histograms, suggesting no general consensus on how easy they were to use. Of the non-head-tracking modes, the Joystick mode is widely considered an easy-to-use steering mode. Spaceball mode and 6-D Mouse mode both tended to be on the difficult side.

The preference rankings show that Joystick mode was widely pre-

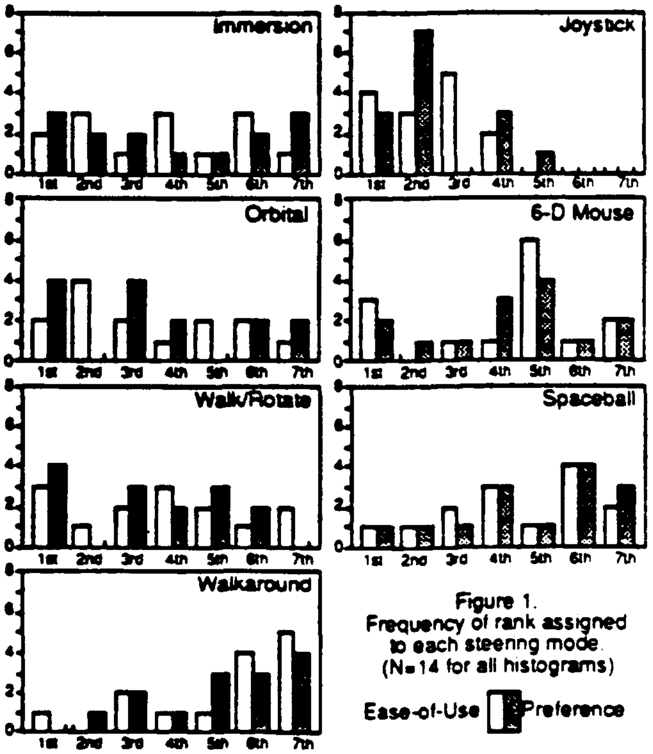


Figure 1.
Frequency of rank assigned to each steering mode.
(N=14 for all histograms)

[*] Spaceball™ is a registered trademark of Spatial Systems, Inc.

[†] 3Space™ is a registered trademark of Polhemus Navigation Sciences.

[‡] EyePhone™ is a registered trademark of VPL Research, Inc.

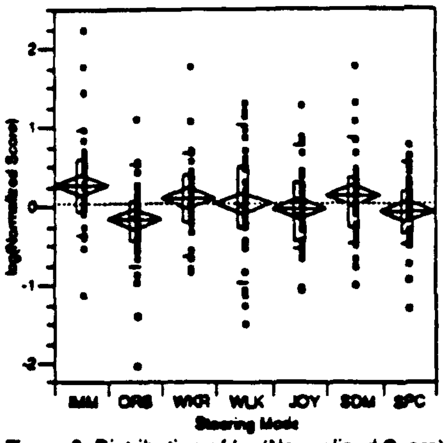


Figure 2. Distribution of log(Normalized Score) by Steering Mode

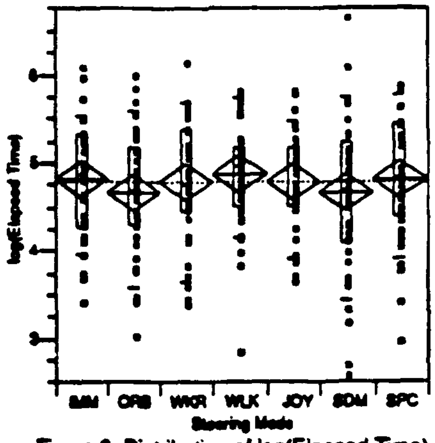


Figure 3. Distribution of log(Elapsed Time) by Steering Mode

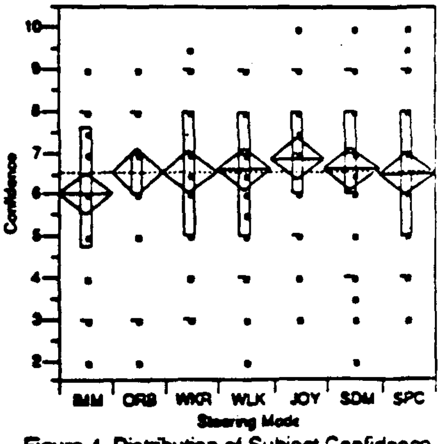


Figure 4. Distribution of Subject Confidence by Steering Mode

Key for Figures 2, 3, and 4

• Data point

Quantiles: 90%, 75%, 50%, 25%, 10%

◇ Group mean with 95% confidence interval

—— Total response sample mean

ferred by the subjects, whereas Walkaround Mode was widely disliked. For Spaceball and 6-D Mouse modes subjects' opinions were on the less-preferred sides. On the other hand, Walk/Rotate and Orbital modes were on the more-preferred side of the scale. Immersion mode shows an interesting bimodal distribution, suggesting that subjects either loved it or hated it.

In order to factor out inter-model variability, each trial's score was normalized by the median score across subjects for the particular model used in that trial. Figure 2 shows the distribution of the logarithms of the normalized scores grouped by steering mode. As the best possible score is 0 (no intersection between beam and dodges), the more negative values represent better performance. Student's t-test reveals significant differences for the following inter-mode comparisons: IMM-ORB (α=0.0007), IMM-SPC (α=0.0069), SDM-ORB (α=0.0126), WKR-ORB (α=0.0187), IMM-JOY (α=0.0197). Head-tracked modes (IMM, ORB, WKR, WLK) taken as a group do not differ significantly from non-head-tracked modes (JOY, SDM, SPC).

Figure 3 shows the distribution of the logarithms of the task completion times grouped by steering mode. No significant differences are found in this data, neither between individual steering modes nor between head-tracked and non-head-tracked modes.

Figure 4 presents the distribution of subject's confidence rating grouped by steering mode. The only significant effect found in this data is the JOY-IMM comparison (α=0.0375).

Table 1 describes correlations between the dependent variables. Not surprisingly, ease-of-use and preference rank are highly correlated. Significant correlations are also found between subject confidence and ease-of-use, preference, and elapsed time. All three correlations are negative, indicating that a subjects' confidence decreased when using difficult steering modes or modes they did not like, or when trials took a long time. Interestingly, score is not significantly correlated with any of the other variables.

| Correl Coeff (Signif. Prob.) | Ease-of-Use Rank | Preference Rank | log(Normalized Score) | log(Elapsed Time) | Confidence |
|---|---|---|---|---|---|
| Ease-of-Use Rank | —<br>(—) | 0.872475<br>(0.0000) | 0.050802<br>(0.3845) | 0.046780<br>(0.4241) | -0.19504<br>(0.0008) |
| Preference Rank | 0.872475<br>(0.0000) | —<br>(—) | 0.008286<br>(0.8740) | 0.07005<br>(0.2311) | -0.22578<br>(0.0001) |
| log(Normalized Score) | 0.050802<br>(0.3845) | 0.008286<br>(0.8740) | —<br>(—) | -0.08843<br>(0.1280) | -0.2238<br>(0.7041) |
| log(Elapsed Time) | 0.046780<br>(0.4241) | 0.07005<br>(0.2311 | -0.08843<br>(0.1280) | —<br>(—) | -0.25238<br>(0.0000) |
| Confidence | -0.19504<br>(0.0008) | -0.22578<br>(0.0001) | -0.2238<br>(0.7041) | -0.25238<br>(0.0000) | —<br>(—) |

Table 1. Correlation coefficients between dependent variables, with significance probability.

## DISCUSSION

### Trial Replay

In addition to the statistical summaries presented above, a subjective review of each trial was conducted by playing back log files in which were recorded status information for the subject's head, the model and the beam at half-second intervals. By observing the trial replay with the HMD, it was possible to study how the subject moved and how the model was manipulated. The trial replay also traced the location of the beam source through the trial, quickly revealing which beam directions were considered, and perhaps more important, which directions were not considered.

In spite of being instructed to find the best possible beam direction, subjects usually terminated the trial before considering all possibilities. Presumably they were able to attain a good enough spatial understanding of the model without having to inspect it from all angles. Trials in which the model had been completely covered usually were usually of extremely long duration, with subject movement suggesting confusion and disorientation. In only a few cases did subjects follow a systematic search strategy, and these systematic searches would usually be abandoned after one candidate beam direction had been found. For the most part, subjects followed what might be called a "greedy" steering strategy, moving about the model in a manner based upon their current view of the model, and not upon some predefined plan. As a result, in most trials the traces of the beam source showed large "holes" that were never considered. From just watching the trial replay it is difficult to determine whether such holes were areas that were deliberately skipped or accidentally missed. Some of these areas corresponded to beam directions that were obviously bad, which might imply that those possibilities were deliberately skipped. Other holes contained prospective beam directions that were good enough to deserve consideration, implying that these areas were accidentally missed by the subject. In most cases the beam directions that required the subject to look straight up or straight down were not covered, as the HMD would exert very large torques on the subject's neck in these positions.

Most subjects relied very heavily on the marker to provide a reference point in the model. The marker served as a landmark that facilitated quick movement between two diametrically opposed beam directions, and was also typically used as a "best-beam-direction-so-far" marker to which the subject would return for the final solution after further exploration elsewhere. Many subjects expressed a desire to have more than one marker. Most subjects did not make use of the context provided by dodges uniquely colored in HLS space for reference. Only one subject, whose own research is concerned with the use of color, found the colors useful—so useful, in fact, that the markers were never used.

### Steering Mode Summaries

*Immersion (IMM).* Immersion mode produced significantly worse scores than Orbital, Spaceball, and Joystick modes, and it appeared to have instilled less confidence in the subjects than the other modes. This may be a result of the subjects' being able to see only a small portion of the model at any time, which, combined with the lack of any head-motion parallax, could have hindered the subject's development of a complete mental picture of the model. In addition, subjects were required to evaluate prospective beam orientations by looking in one direction and then in the other direction, with no clear indication of where the boundary of the beam was. Immersion mode did, however, have the advantage of providing the ability for the subject to use muscle memory in navigation. Even without a complete global understanding of the model, subjects knew how they had to orient their heads to get back to a particular beam direction.

*Orbital (ORB).* Despite the fact that there is no real-world metaphor for this steering mode, Orbital mode produced significantly better scores than Immersion, 6-D Mouse, and Walk/Rotate modes. This may have been due to the unique combination of several factors. Orbital mode provides a beam's-eye view of the model, which at once gives the subject an external global view of the model and allows the subject to easily determine which dodges intersected the beam. Another contributing factor is the aid to navigation through muscle memory provided by Orbital mode.

*Walkaround (WLK).* Walkaround mode produced the longest mean task completion time, but was undistinguished in score and subject confidence. The long task completion time is not surprising, given the difficulty of walking about in the virtual world in a HMD that seals off any view of the real world. Most subjects found this mode very awkward and time-consuming, and ranked Walkaround low in ease of use and preference. Interestingly, this mode more than any other was used for systematic searches. One subject repeatedly circled around the model, inspecting the model at different heights with each loop. Another subject opted to walk less and inspect the model vertically at regular intervals around the model. Perhaps the awkwardness of the mode instilled in these subjects a need for a disciplined, efficient approach.

*Walkaround/Rotation (WKR).* Walk/Rotate mode did not perform any better than Walkaround mode, but fared better in ease and preference rankings. The model rotation capability was used to different degrees by the different subjects. Most subjects walked very little and spent most of their time standing still and rotating the model as in 6-D Mouse mode. Some trials showed no rotation at all, perhaps indicating a reluctance in the subject to lose the navigational advantage provided by a fixed model reference frame.

*Joystick (JOY).* Joystick mode ranked very high in ease-of-use and preference, probably because most of the subjects worked with computers and were somewhat familiar with video games. Even so, performance with Joystick mode was not notable. Trial replay revealed that most subjects used only principal axis rotations, i.e. they rotated models mostly vertically and horizontally and very little diagonally. This was probably due to the mechanical action of the joystick, which required slightly more effort to move diagonally. The effect of this restriction is unclear, for while it forced subjects to decompose their movements into a series of principal axes rotations, it provided a precision of movement not available with the other non-head-tracked modes.

*6-D Mouse (SDM).* Compared to subjects' preference for Joystick mode and dislike for Walkaround mode, response to 6-D Mouse mode was relatively flat. Its performance was undistinguished from the other modes. Trial replays showed that this mode suffered greatly from tracker latency, which greatly hindered both precise alignment and movements large enough to require more than one grab-release cycle. Consequently, beam source traces for 6-D Mouse mode were characterized by a very jagged appearance with large direction changes separating relatively small rotations.

*Spaceball (SPC).* The performance of Spaceball mode is relatively undistinguished, but its preference rankings are weighted toward the low end. Many subjects found the Spaceball fatiguing and difficult to use for precise movements.

### General Comments

Perhaps most compelling is the large inter-subject variance seen this experiment, which may have masked significant differences between steering modes and between the collective head-tracked modes and the non-head-tracked modes. Standardized tests of spatial orientation and spatial visualization [4] may provide a normalizing factor to reduce this variance.

Another interesting observation is the large variation seen in the preference and ease-of-use histograms of the head-tracked modes. There was no general consensus about which of the four modes was the best, although Walkaround was generally considered the worst. This suggests that to be widely accepted, an HMD-based targeting tool should have an adaptable user interface that lets users choose the steering mode they want to use. One must also consider, however, that a task so critical as targeting of treatment beams demands optimal performance. Orbital mode provides better performance than the other three, and will be carried over into the next experiment, involving true, anatomical beam-targeting. Since score is not correlated with mode preference, it is expected that performance of users who do not like Orbital mode will not suffer from having to use it.

## CONCLUSIONS

Collected data show no significant difference between head-tracked steering modes and non-head-tracked steering modes in the performance of an abstract beam-targeting task. Orbital mode provided the best overall performance, Immersion mode the worst. The three non-head-tracked modes were not distinguished by performance.

## ACKNOWLEDGEMENTS

Support for this research was received from:

Defense Advanced Research Projects Agency, Contract No. DAEA 18-90-C-0044.
Digital Equipment Corporation, Research Agreement No. 582.
National Science Foundation, Grant No. CDA-8722752.
Office of Naval Research, Grant No. N00014-86-K-0680.

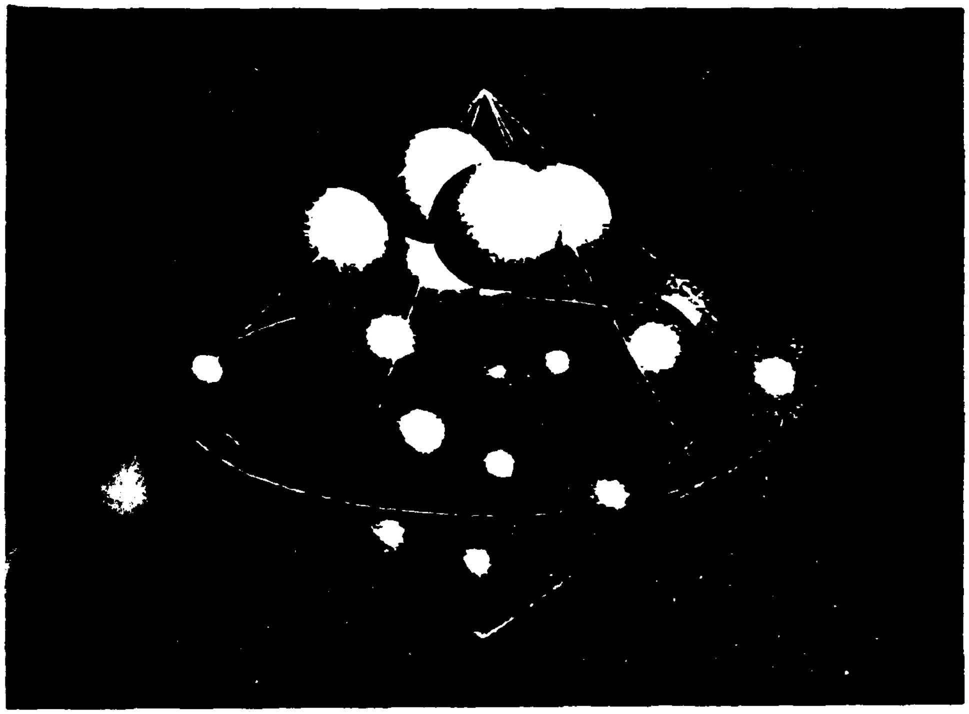
## REFERENCES

1. Ware, C. and Osborne, S., Exploration and virtual camera control in virtual three dimensional environments. In Proc. 1990 Symposium on Interactive 3D Graphics (Snowbird, UT, Mar. '91). Computer Graphics, 24, 2 (Mar. 1991), 175-183.

2. Ware, C. and Slipp, L., Exploring virtual environments using velocity control: A comparison of three devices. In Proc. Hum. Factors Soc., 35th Ann. Mtg. (San Francisco, Sep. '91). HFS, 1991, pp. 300-304.

3. Mackinlay, J.D., Card, S.K., and Robertson, G.G., Rapid controlled movement through a virtual 3D workspace. Computer Graphics, 24, 4 (Aug. 1990), 171-176.

4. McGee, M.G., Human Spatial Abilities. Praeger, New York, 1979.



Color Plate 1. One of the models used in the abstract beam targeting task. The white lines, which were not displayed to the subject, represent the double cone of HLS color space, in which the dodge balls are randomly distributed and colored according to their position in HLS space. The multi-colored target ball is at the center of the double cone.

# Implementation of Flying, Scaling, and Grabbing in Virtual Worlds

Warren Robinett
Richard Hol'ɔway

Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175

## ABSTRACT

In a virtual world viewed with a head-mounted display, the user may wish to perform certain actions under the control of a manual input device. The most important of these actions are flying through the world, scaling the world, and grabbing objects. This paper shows how these actions can be precisely specified with frame-to-frame invariants, and how the code to implement the actions can be derived from the invariants by algebraic manipulation.

## INTRODUCTION

Wearing a Head-Mounted Display (HMD) gives a human user the sensation of being inside a three-dimensional, computer-simulated world. Because the HMD replaces the sights and sounds of the real world with a computer-generated virtual world, this synthesized world is called virtual reality.

The virtual world surrounding the user is defined by a graphics database called a *model*, which gives the colors and coordinates for each of the polygons making up the virtual world. The polygons making up the virtual world are normally grouped into entities called *objects*, each of which has its own location and orientation. The human being wearing the HMD is called the *user*, and also has a location and orientation within the virtual world.

To turn the data in the model into the illusion of a surrounding virtual world, the HMD system requires certain hardware components. The *tracker* measures the position and orientation of the user's head and hand. The *graphics engine* generates the images seen by the user, which are then displayed on the HMD. The *manual input device* allows the user to use gestures of the hand to cause things to happen in the virtual world.

## BASIC ACTIONS

An *action* changes the state of the virtual world or the user's viewpoint within it under control of a gesture of the hand, as measured by the manual input device. The hand gesture initiates and terminates the action, and the changing position and orientation of the hand during the gesture is also used to control what happens as the action progresses.

The manual input device may be a hand-held manipulandum with pushbuttons on it, or it may be an instrumented glove. In either case, the position and orientation of the input device must be measured by the tracker to enable manual control of actions. The input device must also allow the user to signal to the system to start and stop actions, and to select among alternative actions.

Certain fundamental manually-controlled actions may be implemented for any virtual world. These actions involve changing the location, orientation or scale of either an object or a user, as shown in Table 1.

| | User | Object |
|---|---|---|
| Translate | fly through the world | grab (and move) object |
| Rotate | tilt the world | grab (and turn) object |
| Scale | expand or shrink the world | scale object |

Table 1. Basic actions

*Flying* is defined here as an operation of translating in the direction pointed by the hand-held input device, with steering done by changing the hand orientation. This is different from the type of flying available in a flight simulator, where the user can not only translate but can also cause the virtual world to rotate around him by banking. However, translation-only flying is appropriate for a HMD because the user has the ability to turn and look in any direction, and to point the input device in any direction. We believe that keeping the orientation of the virtual world locked to that of the real world helps the user to navigate while flying through the virtual world.

*Tilting the world* is the ability to re-orient the virtual world relative to the user's orientation; that is, to turn the surrounding virtual world sideways. This is implemented by rotating the user with respect to the virtual world, which is subjectively perceived by the user as the entire virtual world rotating around him.

*Scaling the world* is the capability to shrink or expand the world relative to the user, as occurs to Alice in Wonderland when she drinks from the little bottle or eats the little cake. By setting up the action code properly, the user can shrink and expand the world while manually steering the center of expansion. This enables a powerful method of travel in very large virtual worlds: the user shrinks the world down until the destination is within arm's reach and then expands the world, continuously steering the center of expansion so as to arrive at the correctly-scaled destination.

*Grabbing an object* is picking up and moving a simulated object that appears in the virtual world. By analogy with real-world grabbing of objects, this includes the ability to rotate the held object before releasing it.

*Scaling an object* is just shrinking or expanding an individual object alone.

This paper seeks to answer the following question: How can the basic actions of flying, grabbing, scaling and tilting in a HMD system be specified and implemented?

## PRIOR WORK

The first HMD was built in 1968 by Ivan Sutherland [8], but since it had no manual input device other than a keyboard, it did not allow actions controlled by manual gestures. At the University of Utah, a tracked manual input device called a "wand" was added to the system [9]. The tip of the wand was tracked in position but not orientation. The wand was used to deform the surfaces of virtual objects composed of curved patches [2].

In 1985 at NASA Ames Research Center, McGreevy and Humphries built a HMD which was later improved by Fisher, Robinett and others [3]. Under contract to NASA, VPL Research provided an instrumented glove, later named the "DataGlove," which served as a manual input device. The position of the hand and head were tracked with a Polhemus 3Space magnetic tracker. In 1986 using the glove input device, Robinett implemented on this system the actions of flying through the world, scaling the world, rotating the world, and grabbing objects.

Some of these actions, particularly flying and grabbing objects, have since been implemented on HMD systems at several sites. VPL Research began in 1989 selling commercially a HMD system that used a glove to control the actions of flying and grabbing [1]. At the University of North Carolina [7][5], the actions of flying, scaling and grabbing were controlled with a hand-held manual input device with pushbuttons on it which was made from a billiard ball.

## COORDINATE SYSTEMS DIAGRAM FOR A HMD

Various coordinate systems co-exist within a HMD system. All of these coordinate systems exist simultaneously, and although over time they may be moving with respect to one another, at any given moment each pair of them has a relative position and orientation. The instantaneous relationship between two coordinate systems can be described with a transform that converts the coordinates of a point described in one coordinate system to the coordinates that represent that same point in the second coordinate system.

Although transforms exist between any pair of coordinate systems in the HMD system, certain pairs of coordinate systems have relative positions that are either constant, measured by the tracker, or are known for some other reason. These are the *independent transforms*, which are shown in relation to one another in Figure 1. In this diagram, each node stands for a coordinate system, and each edge linking two nodes stands for a transform between those two coordinate systems.

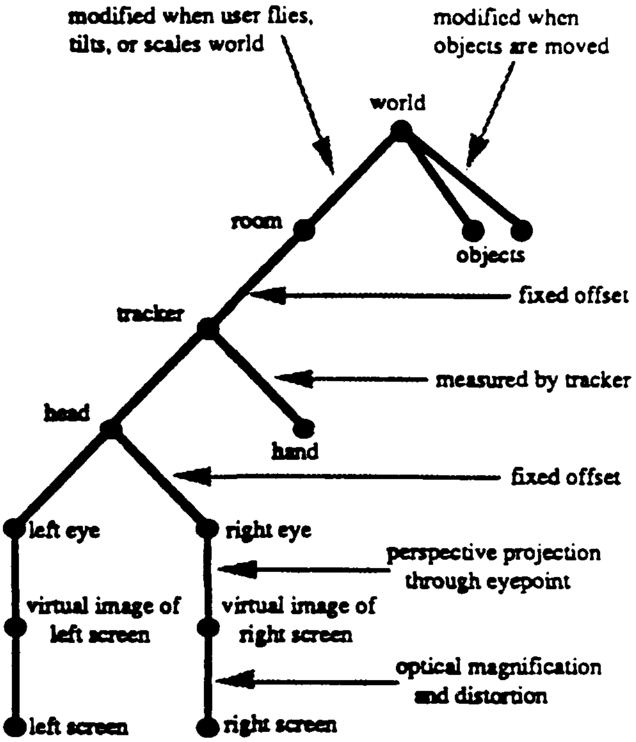

modified when user flies, tilts, or scales world
modified when objects are moved
world
room
objects
fixed offset
tracker
measured by tracker
head
hand
fixed offset
left eye
right eye
perspective projection through eyepoint
virtual image of left screen
virtual image of right screen
optical magnification and distortion
left screen
right screen

Figure 1. Coordinate systems diagram for a single-user HMD system

## NOMENCLATURE FOR TRANSFORMS

We abbreviate the coordinate systems with the first letters of their names. The World-Object transform may be written as $T_{WO}$. Transform $T_{WO}$ converts a point $P_O$ in coordinate system O to a point $P_W$ in coordinate system W.

$$P_W = T_{WO} \cdot P_O$$

This notation is similar to that used in [4]. Notice that the subscripts cancel nicely, as in [6]. Likewise, the composition of the transform $T_{WO}$ going from O to W with the transform $T_{RW}$ going from W to R gives a transform $T_{RO}$ from O to R, with the cancellation rule working here, too:

$$T_{RW} \cdot T_{WO} = T_{RO}$$

The inverse of transform $T_{WO}$ is written $T_{OW}$.

## SPECIFYING ACTIONS WITH INVARIANTS

An action in a virtual world is performed by activating the input device, such as by pushing a button, and then moving the input device to control the action as it progresses. As an example, grabbing a simulated object requires, for each frame while the grab action is in progress, that a new position for the object be computed based on the changing position of the user's hand.

It is possible to precisely define grabbing and other actions with an *invariant*, which is an equation that describes the desired relationship among certain transforms involved in the action. The invariant is typically stated as a relation between certain transforms in the current display frame and certain transforms in the previous frame. In the case of grabbing, the invariant to be maintained is that the Object-Hand transform be equal to its value in the previous frame while the grab action is in progress; in other words, that the object remain fixed with respect to the hand while it is being grabbed.

Starting from the invariant and a diagram of the coordinate systems involved, a mathematical derivation can be performed which produces a formula for updating the proper transform to cause the desired action to occur. For grabbing, this would be updating the Object-World transform to change the object's position and orientation in the virtual world.

Rigorously deriving the update formula from a simple invariant is much easier and more reliable than attempting to write down the update formula using the coordinate systems diagram and informal reasoning. Also, the matching of adjacent subscripts in the notation helps to check that the transforms are in correct order.

## GRABBING AN OBJECT

To derive the update formula for grabbing, we first look at the relevant part of the coordinate system diagram, shown in Figure 2.

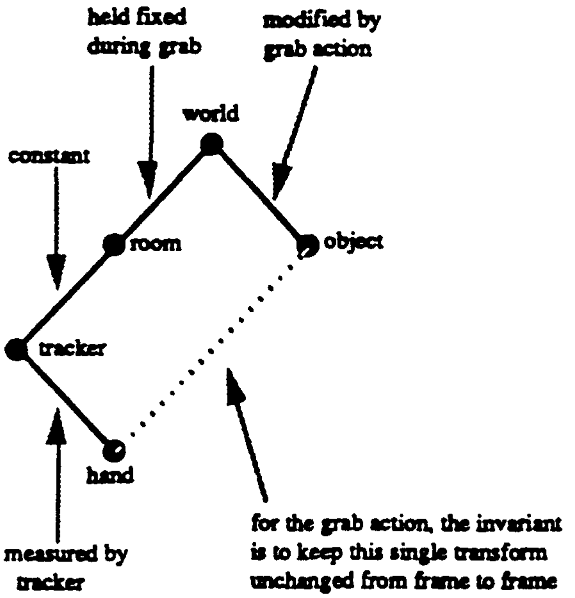


Figure 2. Coordinate systems diagram for grabbing an object

A way of describing the action of grabbing is that the Object-Hand transform $T_{OH}$ remain unchanged from frame to frame, which is expressed by the invariant

$$T_{OH}' = T_{OH}$$

where the apostrophe in $T_{OH}'$ indicates a transform in the current frame which is being updated, and no apostrophe means the value of the transform from the previous frame.

To move an individual object, the Object-World transform $T_{OW}$ must be updated each frame in a way that preserves the invariant. To derive the update formula for grabbing, we start with the invariant and decompose the transforms on both sides based on the relationships among the coordinate systems as shown in the coordinate system diagram.

$$T_{OW}' \cdot T_{WR}' \cdot T_{RT}' \cdot T_{TH}' = T_{OW} \cdot T_{WR} \cdot T_{RT} \cdot T_{TH}$$

We then use algebraic manipulations to isolate the desired transform on the left side of the equation, remembering that these transforms are not commutative.

$$\begin{aligned} T_{OW}' \cdot T_{WR}' \cdot T_{RT}' &= T_{OW} \cdot T_{WR} \cdot T_{RT} \cdot T_{TH} \cdot T_{HT}' \\ T_{OW}' \cdot T_{WR}' &= T_{OW} \cdot T_{WR} \cdot T_{RT} \cdot T_{TH} \cdot T_{HT}' \cdot T_{TR}' \\ T_{OW}' &= T_{OW} \cdot T_{WR} \cdot T_{RT} \cdot T_{TH} \cdot T_{HT}' \cdot T_{TR}' \cdot T_{RW}' \end{aligned}$$

This is the update formula for grabbing, which updates the Object-World transform based on its previous value, the current and previous values of the Hand-Tracker transform (which changes as the hand moves), and the values of the intervening transforms between Tracker and World. The effect of executing this assignment each frame is to keep the object in a fixed position and orientation relative to the hand, even though the hand is moving around within the virtual world.

Another action which can be implemented in a similar manner is "grabbing the fabric of space." In this case, the user can grab and tilt the entire virtual world, rather than just a single object, by holding the World-Hand transform invariant while the hand rotates.

## FLYING

The action of flying is translating the user through the virtual world in the direction pointed by the manual input device. The user steers by rotating the manual input device as the flight proceeds. A metaphor for this type of flying is that the user holds a rocket pistol in his hand, which drags him through the virtual world when he squeezes the trigger.

The manual input device is considered to point in a particular direction that is relative to its local coordinate system. This may be thought of as a 3D vector in Hand coordinates, where the vector's length specifies the flying speed and the vector's direction defines the direction the input device points. This vector defines a translation transform, $T_{H\text{translate}H}$, which moves a point in Hand coordinates to a new position in Hand coordinates. To implement flying, we first need to convert this transformation to operate on points in Room coordinates.

$$T_{R\text{translate}R}' = T_{RH}' \cdot T_{H\text{translate}H}' \cdot T_{HR}'$$

To make the user's position change within the virtual world, the World-Room transform must be modified each frame, so the invariant for flying is

$$T_{WR}' = T_{WR} \cdot T_{R\text{translate}R}'$$

which may be expanded to give the update formula for flying.

$$T_{WR}' = T_{WR} \cdot T_{RT}' \cdot T_{TH}' \cdot T_{H\text{translate}H} \cdot T_{HT}' \cdot T_{TR}'$$

## SCALING THE WORLD

It is possible to shrink or expand the surrounding virtual world. This is comprehensible and effective because the user has direct perception of the size of and distance to virtual objects through stereopsis and head-motion parallax, and can therefore easily perceive the concerted motions of the objects in the virtual world expanding around a center of expansion, or shrinking towards a center of contraction.

The type of scaling used is uniform scaling, in which all three dimensions are always scaled by the same factor. There is always a center of scaling when uniform scaling occurs, and for the manually controlled action of scaling the world, it makes sense to locate the center of scaling at the user's hand. When expanding the world, the center of scaling is the point that virtual objects move away from as expansion occurs, and so to end up at a specific desired location within a formerly-tiny virtual world, the center of scaling must be repeatedly re-centered on the desired location as it emerges during expansion.

Implementing this action requires a derivation similar to that used for flying. An incremental scaling transformation in Hand coordinates,$T_{HscaleH}$, will use the Hand origin as the center of scaling. Below we give the invariant for scaling the world, and the update formula derived from it.

$$T_{WR}' = T_{WR} \cdot T_{RscaleR}'$$
$$T_{WR}' = T_{WR} \cdot T_{RT}' \cdot T_{TH}' \cdot T_{HscaleH} \cdot T_{HT}' \cdot T_{TR}'$$

## GENERAL FORM

Upon examining the invariants for flying and scaling, we see a strong similarity between them: both invariants are of the form:

$$T_{WR}' = T_{WR} \cdot T_{R<transform>R}'$$

In fact, these two invariants for updating $T_{WR}$ are examples of a more general technique for updating a transform between two coordinate systems based on a transform that occurs in a third coordinate system. The general form for updating the transform $T_{AB}$ in terms of an action in coordinate system K is:

$$T_{AB}' = T_{AB} \cdot T_{BK}' \cdot T_{K<transform>K} \cdot T_{KB}'$$

where there may be an arbitrary number of coordinate systems between B and K, and $T_{BK}$ is the product of the transforms that go between the two coordinate systems.

Using this general form, scaling an object about the hand is analogous to scaling the world about the hand:

$$T_{OW}' = T_{OW} \cdot T_{WH}' \cdot T_{HscaleH} \cdot T_{HW}'$$

## CONCLUSIONS

The foregoing examples of grabbing, flying and scaling show how actions can be implemented that operate under continuous manual control by the user. For each action, the relationship between the motion of the hand and the transforms to be modified was precisely specified with an invariant. These invariants not only provided a concise and precise specification of each action, but also provided a starting point for a formal derivation that produced update equations which could be used directly to implement the actions.

Using invariants and derivations to produce the code to implement grabbing, scaling and flying is greatly superior to the method which is often used, namely, to just write down a sequence of transforms that looks right based on the coordinate system diagram. It is easy to get some of the transforms in the wrong order. The notation used in this paper provides a check against misordering the transforms by requiring adjacent subscripts to match. The HMD software at UNC was implemented using this notation and the formulas derived in this paper, and serves as proof that they work.

## ACKNOWLEDGEMENTS

We would like to thank many people for their contributions to this work, starting with the HMD and Pixel-Planes teams at UNC, led by Fred Brooks and Henry Fuchs. We thank Fred Brooks for useful discussions about coordinate systems diagrams and nomenclature. This work builds on earlier work at NASA, and we would like to acknowledge the contributions of Scott Fisher, Jim Humphries, Doug Kerr and Mike McGreevy. We thank Ken Shoemake for help with quaternions and Julius Smith for rules-of-thumb about writing technical papers. This research was supported by the following grants: DARPA #DAEA 18-90-C-0044, NSF Cooperative Agreement #ASC-8920219, and DARPA: "Science and Technology Center for Computer Graphics and Scientific Visualization", ONR #N00014-86-K-0680, and NIH #5-R24-RR-02170.

## REFERENCES

[1] Blanchard, C., S. Burgess, Y. Harvill, J. Lanier, A. Lasko, M. Oberman, M. Teitel. Reality Built for Two: A Virtual Reality Tool. *Proc. 1990 Workshop on Interactive 3D Graphics*. 35-36.

[2] Clark, J. 1976. Designing surfaces in 3-D. *Communications of the ACM*. 19:8:454-460.

[3] Fisher, S., M. McGreevy, J. Humphries, and W. Robinett. 1986. Virtual Environment Display System. *Proc. 1986 Workshop on Interactive 3D Graphics*. 77-87.

[4] Foley, J., A. van Dam, S. Feiner, J. Hughes. 1990. *Computer Graphics: Principles and Practice* (2nd ed.). Addison-Wesley Publishing Co., Reading MA. 222-226.

[5] Holloway, R., H. Fuchs, W. Robinett. 1991. Virtual-Worlds Research at the University of North Carolina at Chapel Hill. *Proc. Computer Graphics '91*. London, England.

[6] Pique, M. 1980. Nested Dynamic Rotations for Computer Graphics. M.S. Thesis, University of North Carolina, Chapel Hill, NC.

[7] Robinett, W. 1990. Artificial Reality at UNC Chapel Hill. [videotape] *SIGGRAPH Video Review*.

[8] Sutherland, I. 1968. A head-mounted three-dimensional display. *1968 Fall Joint Computer Conference, AFIPS Conference Proceedings*. 33:757-764.

[9] Vickers, D. 1974. Sorceror's Apprentice: head mounted display and wand. Ph.D. dissertation, Dept. of Computer Science, Univ. of Utah, Salt Lake City.

# A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems

Mark Ward†, Ronald Azuma, Robert Bennett, Stefan Gottschalk, Henry Fuchs

Department of Computer Science
Sitterson Hall
University of North Carolina
Chapel Hill, NC 27599-3175

## Abstract

An optoelectronic head-tracking system for head-mounted displays is described. The system features a scalable work area that currently measures 10' x 12', a measurement update rate of 20-100 Hz with 20-60 ms of delay, and a resolution specification of 2 mm and 0.2 degrees. The sensors consist of four head-mounted imaging devices that view infrared light-emitting diodes (LEDs) mounted in a 10' x 12' grid of modular 2' x 2' suspended ceiling panels. Photogrammetric techniques allow the head's location to be expressed as a function of the known LED positions and their projected images on the sensors. The work area is scaled by simply adding panels to the ceiling's grid. Discontinuities that occurred when changing working sets of LEDs were reduced by carefully managing all error sources, including LED placement tolerances, and by adopting an overdetermined mathematical model for the computation of head position: space resection by collinearity. The working system was demonstrated in the Tomorrow's Realities gallery at the ACM SIGGRAPH '91 conference.

**CR categories and subject descriptors:** I.3.1 [Computer Graphics]: Hardware Architecture - *three-dimensional displays*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *Virtual Reality*

**Additional Key Words and Phrases:** Head-mounted displays, head tracking

## 1 Introduction

It is generally accepted that deficiencies in accuracy, resolution, update rate, and lag in the measurement of head position can adversely affect the overall performance of a HMD [17][24][25]. Our experience suggests that an additional specification requires more emphasis: range.

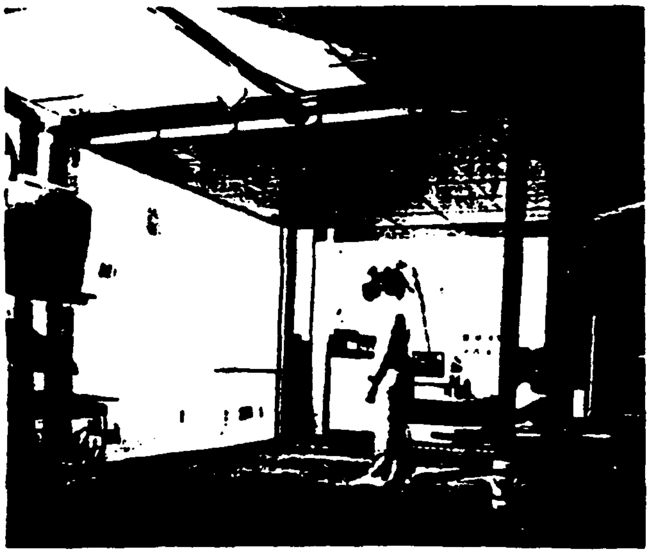


*Figure 1: The existing system in UNC's graphics laboratory*

Most existing HMD trackers were built to support situations that do not require long-range tracking, such as cockpit-like environments where the user is confined to a seat and the range of head motion is limited. But many virtual worlds applications, such as architectural walkthroughs, would benefit from more freedom of movement (Figure 2). Long-range trackers would allow greater areas to be explored naturally, on foot, reducing the need to resort to techniques such as flying or walking on treadmills.

Such techniques of extending range work adequately with closed-view HMDs that completely obscure reality. With see-through HMDs [9][11], however, the user's visual connection with reality is intact and hybrid applications are possible where physical objects and computer-generated images coexist. In this situation, flying though the model is meaningless. The model is registered to the physical world and one's relationship to both must change simultaneously.

This paper describes the second generation of an optoelectronic head-tracking concept developed at the University of North Carolina at Chapel Hill. In the concept's first generation, the fundamental design parameters were explored and a bench-top prototype was constructed [28]. Building on this success, the second-generation tracker is a

† Present address: Structural Acoustics, 5801 Lease Lane,
Raleigh, NC, 27613. (919) 787-0887

fully functional prototype that significantly extends the workspace of an HMD wearer.

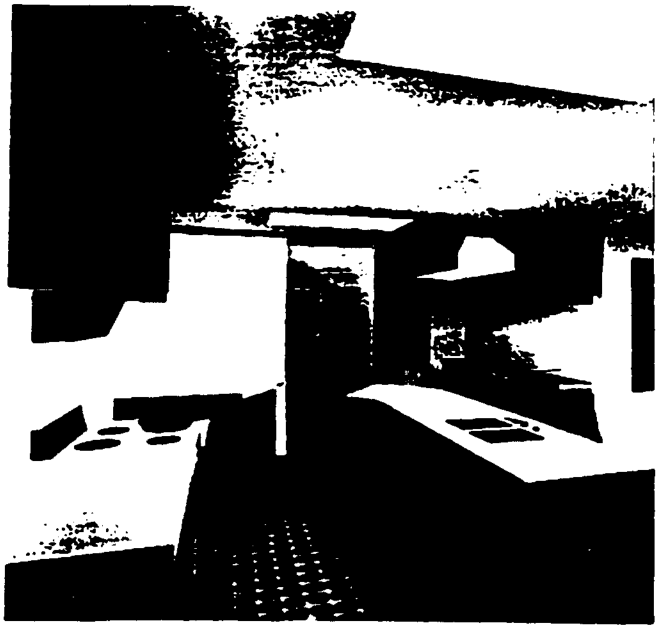


*Figure 2: Walkthrough of Brooks' kitchen design that runs with the tracker. Actual resolution of images seen in the HMD is much lower than this picture's resolution.*

The current system (Figure 1) places four outward-looking image sensors on the wearer's head and locates LEDs in a 10' x 12' suspended ceiling structure of modular 2' x 2' ceiling panels. Each panel houses 32 LEDs, for a total of 960 LEDs in the ceiling. Images of LEDs are formed by lateral-effect photodiode detectors within each head-mounted sensor. The location of each LED's image on a detector, or *photocoordinate*, is used along with the known LED locations in the ceiling to compute the head's position and orientation. To enhance resolution, the field of view of each sensor is narrow. Thus, as shown in Figures 3 and 7, each sensor sees only a small number of LEDs at any instant. As the user moves about, the working set of visible LEDs changes, making this a *cellular* head-tracking system.

Measurements of head position and orientation are produced at a rate of 20-100 Hz with 20-60 ms of delay. The system's accuracy has not been measured precisely, but the resolution is 2 mm and 0.2 degrees. It was demonstrated in the Tomorrow's Realities gallery at the ACM SIGGRAPH '91 conference, and is, to our knowledge, the first demonstrated scalable head-tracking system for HMDs.

The system is novel for two reasons. First, the sensor configuration is unique. Other optical tracking systems fix the sensors in the environment and mount the LEDs on the moving body [30]. The outward-looking configuration is superior for it improves the system's ability to detect head rotation. The scalable work space is the system's second contribution. If a larger work space is desired, more panels can be easily added to the overhead grid.

## 2 Previous work

Many tracking systems precede this effort, and we will briefly survey representative examples. The essence of the problem is the realtime measurement of the position and orientation of a rigid moving body with respect to an absolute reference frame, a six-degree-of-freedom (6DOF) measurement problem. Solutions are relevant to many other fields.

To our knowledge, four fundamentally different technologies have been used to track HMDs: mechanical, magnetic, ultrasonic, and optical.

The first HMD, built by Ivan Sutherland [27], used a mechanical linkage to measure head position. A commercial product, The Boom [12], uses a mechanical linkage to measure the gaze direction of a hand-held binocular display. The Air Force Human Resources Laboratory (AFHRL) uses a mechanical linkage to measure the position and orientation of a HMD used for simulation [24]. Mechanical systems have sufficient accuracy, resolution, and frequency response, yet their range is severely limited, and a mechanical tether is undesirable for many applications.

Magnetic-based systems [3][21] are the most widely used hand and head trackers today. They are small, relatively inexpensive, and do not have line-of-sight restrictions. Their primary limitations are distortions caused by metal or electromagnetic fields, and limited range [13].

Ultrasonic approaches have also been successful, such as the commercially-available Logitech tracker [20]. Time-of-flight measurements are used to triangulate the positions of sensors mounted on the HMD. The strength of this technology is minimum helmet weight [13]. Physical obscuration as well as reflections and variations of the speed of sound due to changes in the ambient air density make it difficult to maintain accuracy [5].

Because of the potential for operation over greater distances, optical approaches are plentiful, and it is helpful to categorize them on the basis of the light source used. Visible, infrared, and laser light sources have each been exploited.

Ferrin [13] reports the existence of a prototype helmet tracking system using visible light. Although it only tracks orientation, it is worth mentioning here because of its unique approach. A patterned target is placed on the helmet and a cockpit-mounted video camera acquires images in real time. The pattern is designed to produce a unique image for any possible head orientation. The strength of this approach is the use of passive targets which minimize helmet weight. Reflections and other light sources are potential sources of error.

Bishop's Self-Tracker [7] is a research effort involving visible light. A Self-Tracker chip senses incremental displacements and rotations by imaging an unstructured scene. A head-mounted cluster of these chips provide sufficient information for the computation of head position and orientation. Although still under development, the concept is mentioned here because it would allow an optical tracking system to operate outdoors, where a structured environment, such as our ceiling of LEDs, would be impossible to realize.

Because of the difficulties associated with processing information in an unstructured scene, most high-speed optical measurement systems use highly-structured infrared or laser light sources in conjunction with solid-state sensors. The sensor is a often a lateral-effect photodiode as opposed to a true imaging device, because the photodiode produces currents that are directly related to the location of a light spot's centroid on its sensitive surface [32]. The resultant sensor is relatively insensitive to focus, and the light spot's location, or photocoordinate, is immediately available without the need for image processing.

During the 1970's, Selspot [23] popularized the use of infrared LEDs as targets and lateral-effect photodiodes as sensors in a commercially-available system. Their primary emphasis was, and still is, on the three-dimensional locations of individual targets. That is, the Selspot system does not automate the computation of a rigid body's orientation. In a response to this shortcoming, Antonsson [2] refined the Selspot system for use in dynamic measurements of mechanical systems. The resultant system uses two Selspot cameras to view a moving body instrumented with LEDs. Similar approaches have been applied to HMD systems in cockpits [13] and in simulators [11].

The use of an LED light source limits the range of these systems. Typically, the distance between source and detector can be no greater than several feet. Longer distances can be spanned with laser light sources.

The only known example of a 6DOF tracker using laser sources is the Minnesota Scanner [26]. With this sytem, scanning mirrors are used to sweep orthogonal stripes of light across the working volume. Photodiodes are both fixed in space and placed on the moving body. By measuring the time between a light stripe's contact with a fixed and moving photodiode, the diode's three-dimensional location can be computed. Given the location of three or more moving diodes, the moving body's orientation can be computed. Similar technology has been applied to the cockpit, although orientation was the only concern [13].

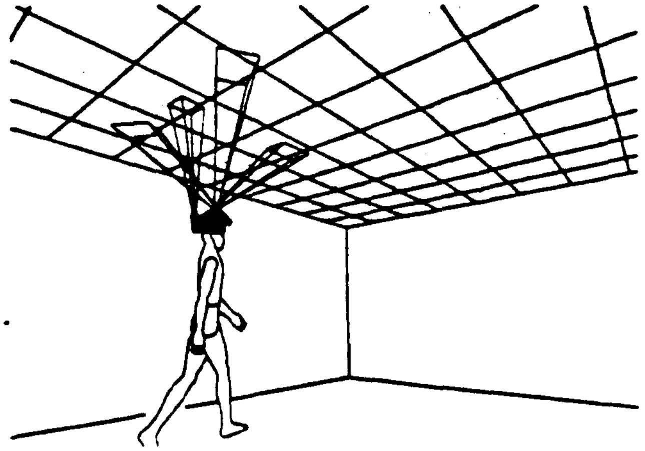


*Figure 3: Conceptual drawing of outward-looking system and the sensors' fields of view*

## 3 System overview

Wang demonstrated the viability of head-mounted lateral-effect photodiodes and overhead LEDs. This system extends his work in several ways. First, an overhead grid of 960 LEDs was produced with well-controlled LED location tolerances, and more attention was paid to controlling other error sources as well. Second, mathematical techniques were developed that allow an arbitrary number of sensors and an arbitrary number of LEDs in the field of view of each sensor to be used in the computation of head location. This resulted in an overdetermined system of equations which, when solved, was less susceptible to system error sources than the previous mathematical approach [10]. Third, the analog signals emerging from the sensors were digitally processed to reject ambient light. Finally, techniques for quickly determining the working sets of LEDs were developed.

### 3.1 Sensor configuration

Typically, optical trackers are *inward-looking*; sensors are fixed in the environment within which the HMD wearer moves. With Self-Tracker, Bishop and Fuchs introduced the concept of *outward-looking* trackers that mount the image sensors on the head, looking out at the environment (Figure 3).

If a large work area is required, outward-looking configurations have an advantage over inward-looking techniques when recovering orientation. The two are equivalent for measuring translation: moving the sensor causes the same image shift as moving the scene. Rotations are significantly different. Unless targets are mounted on antlers, an inward-looking sensor perceives a small image shift when the user performs a small head rotation. The same head rotation creates a much larger image shift with a head-mounted sensor. For a given sensor resolution, an outward-looking system is more sensitive to orientation changes.

*Figure 4: Remote Processor and head unit with four sensors*

To improve resolution in general, long focal lengths must be used with an optical sensor regardless of whether the configuration is inward or outward-looking. Thus, a wide-angle lens cannot significantly extend the work area of an inward-looking system without sacrificing resolution and accuracy.

Narrow fields of view are a consequence of long focal lengths. Therefore, the HMD wearer cannot move very far before an LED leaves a given sensor's field of view. One solution is a cellular array of either LEDs or detectors. For an infrared system using LEDs and lateral-effect photodiodes, system cost is minimized by replicating LEDs as opposed to sensors. This is a result of both the device cost as well as the required support circuitry.

In the current system, four Hamamatsu (model S1880) sensors are mounted atop the head, as shown in Figure 4. Each sensor consists of a camera body to which a Fujinon lens (model CF 50B) is attached. The focal length of each lens is 50mm. Their principal points were determined experimentally by an optical laboratory. An infrared filter (Tiffen 87) is used to reject ambient light.

### 3.2 Beacon configuration

Experience with simulations and an early 48-LED prototype revealed the problem of *beacon switching error*: as the user moved around and the working set of beacons changed, discontinuous jumps in position and orientation occurred. These are caused by errors in the sensor locations, distortions caused by the lens and photodiode detector, and errors in the positions of the beacons in the ceiling.

To control beacon locations, we housed the LEDs in carefully constructed ceiling panels. Each 2' x 2' panel is an anodized aluminum enclosure that encases a 20" x 20" two-sided printed circuit board. On this board are electronics to drive 32 LEDs. The LEDs are mounted in the front surface with standard plastic insets. Using standard electronic enclosure manufacturing techniques, it was relatively easy to realize an LED-to-LED centerline spacing tolerance of .005" on a given panel.

The panels are hung from a Unistrut superstructure (Figure 1). At each interior vertex of a 2' x 2' grid, a vertically adjustable hanger mates with four panels. Four holes in the face of a panel slide onto one of four dowels on each hanger. The entire array of panels is levelled with a Spectra Physics Laser-Level, which establishes a plane of visible red light several inches below the panels' faces. Each hanger is designed to accept a sensor (Industra-Eye) that measures the vertical position of the laser relative to its own case. By moving the hangers up or down, they can be aligned to within .006" of the light beam.

The panels are electrically connected by a data and power daisy chain. The data daisy chain allows an individual LED to be selected. Once selected, the LED (Siemens SFH 487P) can be driven with a programmable current that ranges from 0-2 amperes. The programmable current allows an electronic iris feature to be implemented. Typically, an LED will be on for no more than 200 µsec. During this time period, the current is adjusted to achieve a desired signal level at the sensor (see Section 4).

### 3.3 Data Flow

As shown in Figure 5, the signals emerging from the head-mounted sensors are connected to the Remote Processor. Worn as a belt pack, the Remote Processor functions as a remote analog-to-digital conversion module. It can accept the four analog voltages emerging from a lateral-effect photodiode, for up to eight sensors. On command, the Remote Processor will simultaneously sample the four voltages on a selected sensor and relay four, 12-bit results to the LED Manager. The Remote Processor was used to alleviate the need for long runs of analog signals emerging from multiple sensors.

The LED Manager is a 68030-based processing module that controls the Remote Processor as well as the ceiling. A TAXI-based serial datalink [1] provides access to the Remote Processor while the ceiling's data daisy chain terminates at the LED Manager. Software executing on this module is responsible for turning LEDs on and for extracting data from the sensors. The LED Manager resides in a remote VME chassis that must be located near the ceiling structure.

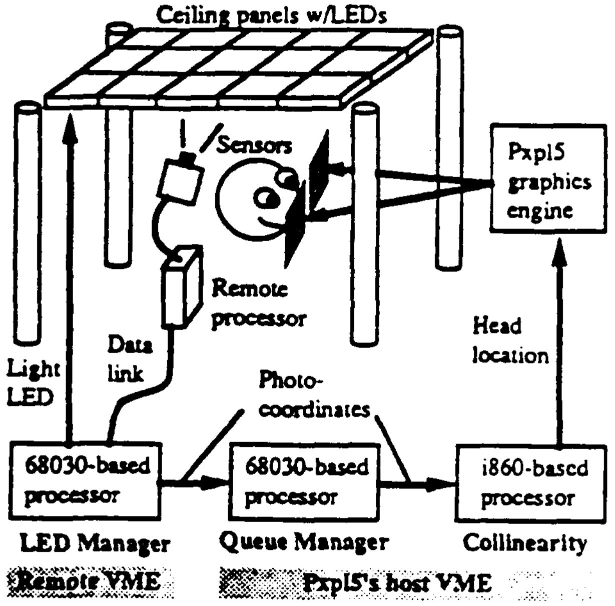


*Figure 5: System Dataflow*

For each measurement of head location, the LED Manager produces a list of visible LEDs and their associated photocoordinates. This list is transferred via shared memory to the Collinearity module, which resides in the graphics engine's VME chassis. The i860-based Collinearity module translates the list of photocoordinates into the current estimate of head location. For reasons explained in Section 6, an additional 68030-based processor is used to aid the transfer of data from the remote system to the host. In theory, this is not required. The VME systems are connected by a Bit-3 VME buslink.

The sampled head position is communicated to the Pixel-Planes 5 graphics engine [14], which in turn updates the images on the user's displays.

## 4 Low-level software

A library of low-level routines running on the LED Manager, called the Acquisition Manager, controls the beacons and detectors. Given an LED and a photodiode unit, these routines light an LED and determine if a photodiode's detector sees that LED. The detector returns four analog signals, which the Remote Processor board digitizes. A simple formula [16] converts these four numbers into the $x,y$ photocoordinates of the LED's projection on the detector.

Hamamatsu datasheets specify 1 part in 40 accuracy and 1 part in 5000 resolution for the lateral-effect diode-based detectors used. As with Antonsson [2], we were able to achieve approximately 1 part in 1000 accuracy for the combined photodiode-lens assembly. Achieving this result required significant efforts to improve the signal-to-noise ratio and compensate for distortion, including:

*Ambient light rejection:* The voltage values with the LED off (called the "dark current") are subtracted from the voltage values with the LED on. Sampling with the LED off both before and after the samples with the LED on and averaging the two yields substantially improved ambient light rejection.

*Random noise rejection:* Averaging several measurements reduces random noise effects, but costs time. A good compromise between accuracy and sampling speed is to take 8 samples with the LED off, 16 samples with the LED on and 8 more samples with the LED off.

*Current scaling:* The distance between a photodiode and an LED depends on the user's location. To maximize the signal without saturating the photodiode detector, the Acquisition Manager dynamically adjusts the amount of current used to light an LED. Acquisition Manager routines estimate the threshold of current that will saturate the detector and use 90% of this value during sampling.

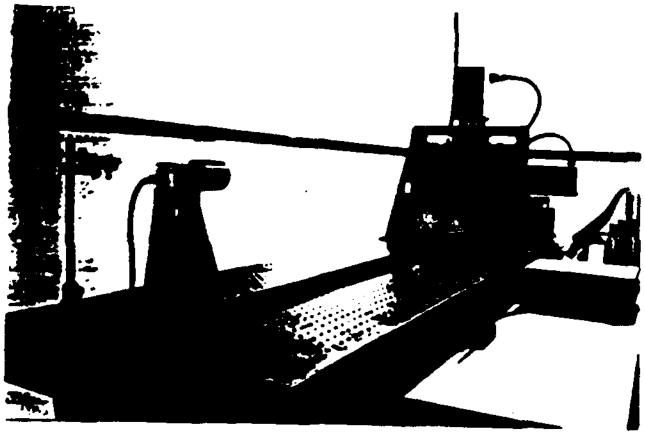


*Figure 6: Optical bench for photodiode calibration*

*Calibration:* Both the lens and the photodiode detector suffer from nonlinear distortions. By placing the photodiodes on an optical bench and carefully measuring the imaged points generated by beacons at known locations (Figure 6), we built a lookup table to compensate for these distortions. Bilinear interpolation provides complete coverage across the detector. More sophisticated calibration techniques should be investigated. Accurate calibration is required to reduce beacon switching error.

*Programming techniques:* Techniques such as list processing, cache management and efficient code sequencing result in a substantially improved sampling rate. In addition, expedited handling of special cases, such as when an LED is not within the field of view of a photodiode unit, further helps system performance.

Using 32 samples per LED, we compute a visible LED's photocoordinate in 660 μsec and reject a non-visible LED in 100 μsec. LEDs are tested in groups; each group carries an additional overhead of 60 μsec.

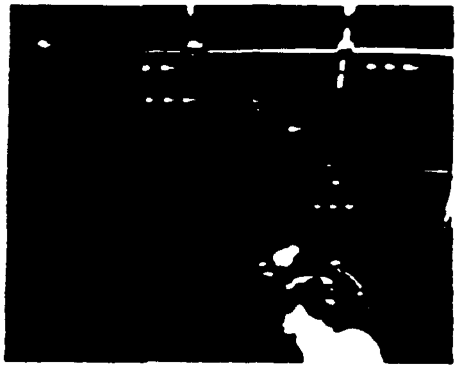


*Figure 7: Sensors viewing LEDs in the ceiling. Each of the four groups is the set of LEDs that a sensor can see. Picture taken with a camera that is sensitive to infrared light.*

## 5 LED Manager

The LED Manager uses the low-level Acquisition Manager routines to determine which LEDs each photodiode unit sees and where the associated imaged points are on the photodiode detectors. We usually want to collect data from all visible LEDs, since larger sample sets ultimately yield less noisy solutions from the Collinearity module (Section 7). Because the number of visible LEDs is small (see Figure 7) compared to the total number of LEDs in the ceiling, something faster than a brute-force scan of the entire ceiling array is called for. Two assumptions help us design a more efficient method:

*1) Spatial coherence:* The set of beacons visible to a photodiode unit in a given frame will be contiguous.

*2) Temporal coherence:* The user's movement rate will be slow compared to the frame rate. This implies that the field of view of a given photodiode unit does not travel very far across the ceiling between frames, so its set of visible beacons will not change much from one frame to the next.

### 5.1 The basic method

In each frame, the LED Manager goes through each photodiode unit in sequence, sampling beacons until it is satisfied that it has captured most of each photodiode unit's visible set. A basic difficulty is that we cannot be sure whether a beacon is visible or not until we attempt to sample it. The LED Manager remembers which beacons were in the camera's visible set from the previous frame. The set is called the *last visible set*. If the last visible set is nonempty, all beacons in that set are tested. The next action depends on how many of those beacons are actually visible:

*1) All:* We assume the field of view has not moved much and not many more beacons will be visible. We stop with this set and go on to the next photodiode unit.

*2) Some:* We assume that the field of view has shifted significantly, possibly enough to include previously unseen beacons. A *shell fill* (described later) is conducted, beginning with the set of beacons verified to be visible.

*3) None:* The field of view has moved dramatically, gone off the edge of the ceiling, or is obscured. We check the neighbors of the last visible set. If any of these beacons are visible, they are used to start a shell fill. If none are visible, we give up on this photodiode unit until the next frame.

What if the last visible set is empty? Our course of action depends on whether we were able to compute a valid position and orientation for the head in the last frame:

*1) Valid previous location:* We can predict which LEDs should be visible to our photodiode unit, if the user's head is actually at the computed location, because the geometry of the head unit is known. If no LEDs are predicted to be visible, we go on to the next photodiode unit, otherwise we sample those beacons and use them as the start of a shell fill, if any of them were actually visible.

*2) No valid previous location:* Now we have no way to guess which beacons are visible, so we resort to a simple *sweep search*, which lights the beacons in the ceiling row by row, until we have tried the entire ceiling or an LED is found to be visible. In the former case, we give up, and in the latter case, we use the visible beacon as the start of a shell fill.

### 5.2 Shell fill

A *shell fill* starts with a set of beacons known to be visible to a sensor and sweeps outward until it has found all the beacons in the field of view.

We do this by first sampling the neighbors of the initial set of beacons. If none are found visible, the shell fill terminates, concluding that the beacons in the initial set are the only visible ones. If any are found visible, we then compute the neighbors of the beacons we just sampled, excluding those which have already been tried, and sample those. We repeat this process of sampling beacons, computing the neighbors of those found visible, and using those neighbors as the next sample set, until an iteration yields no additional visible beacons.

Assumption 1, that visible sets are contiguous, suggests that this procedure should be thorough and reasonably efficient.

### 5.3 Startup

At startup, the head location is not known and all of the last visible sets are empty. We do a sweep search, as previously described, for each photodiode unit to locate the initial visible sets.

## 6 Communications

Communication between the various processors in our system is done using shared memory buffers, which offer low latency and high speed. The buffers are allocated and deallocated via a FIFO queue mechanism. Data is "transmitted" when it is written to the buffer: no copying is necessary. The only communication overhead is the execution of a simple semaphore acquisition and pointer management routine. Furthermore, all processors use the same byte ordering and data type size, so no data translation is needed.

The queuing mechanism lets all modules in the system run asynchronously. LED Manager, the Collinearity module, and Pixel-Planes 5 run as fast as they can, using the most recent data in the queue or the last known data if the queue is empty.

The various processors in our system are split between two separate VME buses, which are transparently linked together by Bit-3 bus link adapters (Figure 5). A subtle bus loading problem prevents the i860 board and the '030 board that runs LED Manager from operating in the same VME cage. This configuration increases latency because inter-bus access is significantly slower than intra-bus access, but increases throughput because the bus link allows simultaneous intra-bus activity to occur. Because the i860 processor cannot directly access the VME bus, a second '030 board, which runs the Queue Manager, moves data between the LED Manager and the Collinearity module.

A simpler and less expensive system could be built if we acquired an i860 board that can run on the same bus as the LED Manager '030 board. This configuration would not require the Queue Manager board or the Bit-3 links and would reduce both latency and throughput.

## 7 Space Resection by Collinearity

Given the observations of beacons, we compute the position and orientation of the user's head by using a photogrammetric technique called space resection by collinearity. The basic method for a single camera is in [31]; what we describe here is our extension for using it in a multi-sensor system. Because of space limitations, the description is necessarily brief. Full details are provided in [6].

### 7.1 Definitions

Three types of coordinate systems exist: one World space (tied to the ceiling structure), one Head space (tied to the HMD), and several Photodiode spaces (one for each photodiode unit).

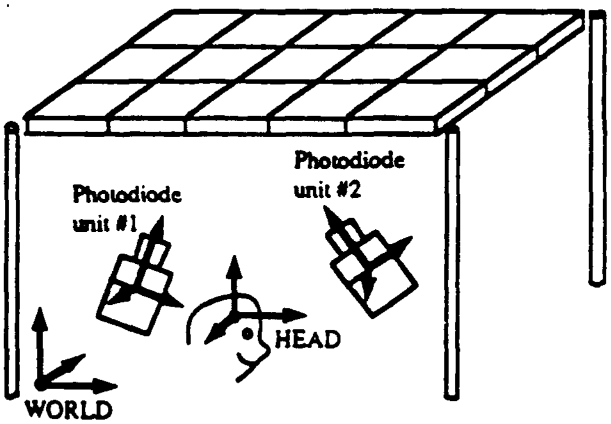


*Figure 8: World, Head and Photodiode spaces*

Changing representations from one space to another is done by a rotation followed by a translation. We use two types of 3x3 rotation matrices:

$M$ = Head space to World space
$M_i$ = Photodiode space $i$ to Head space

with each matrix specified by Euler angles $\omega$, $\alpha$, and $\kappa$.

The optical model for each photodiode unit is simple: a light ray strikes the front principal point and leaves the rear principal point at the same angle (Figure 9).

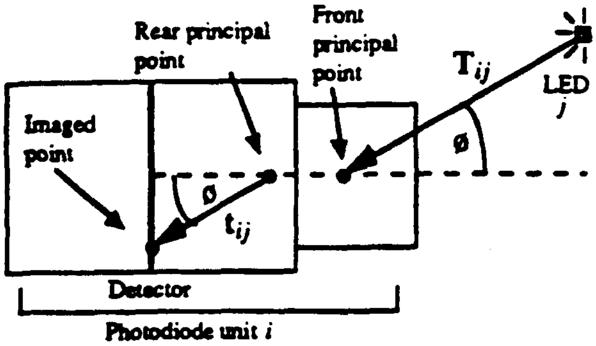


*Figure 9: Optical model*

Finally, we list the points and vectors we will need, segregated by the coordinate system in which they are represented. Given photodiode unit $i$ sees LED number $j$.

Photodiode space:
- $[x_{ij}, y_{ij}, 0]$ = imaged point on photodiode detector

Head space:
- $\mathbf{t}_{ij}$ = vector from rear principal point to imaged point
- $\mathbf{H}_0$ = origin of Head space
- $\mathbf{d}_i$ = vector from $\mathbf{H}_0$ to center of photodiode detector
- $\mathbf{e}_i$ = vector from $\mathbf{H}_0$ to rear principal point
- $\mathbf{f}_i$ = vector from $\mathbf{H}_0$ to front principal point

World space:
- $[X_0, Y_0, Z_0]$ = coordinates of the origin of Head space
- $[X_j, Y_j, Z_j]$ = coordinates of LED $j$
- $\mathbf{T}_{ij}$ = vector from LED $j$ to front principal point

### 7.2 Geometric relationships

Figure 9 shows that $\mathbf{T}_{ij}$ and $\mathbf{t}_{ij}$ differ only by a scale factor; if they were placed at the same start point, they would be collinear. In equations:

$$\mathbf{T}_{ij} = \lambda \, \mathbf{M} \, \mathbf{t}_{ij} \qquad (1)$$

We now express $\mathbf{T}_{ij}$ and $\mathbf{t}_{ij}$ in terms of the other vectors in equations (2) and (3) and Figures 10 and 11:

$$\mathbf{T}_{ij} = \begin{bmatrix} X_0 - X_j \\ Y_0 - Y_j \\ Z_0 - Z_j \end{bmatrix} + \mathbf{M}\,\mathbf{f}_i \qquad (2)$$

$$\mathbf{t}_{ij} = \mathbf{d}_i - \mathbf{e}_i + \mathbf{M}_i \begin{bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{bmatrix} \qquad (3)$$

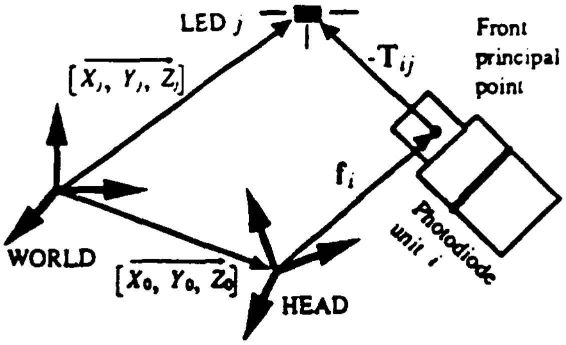


*Figure 10: Expressing $T_{ij}$ through other vectors*

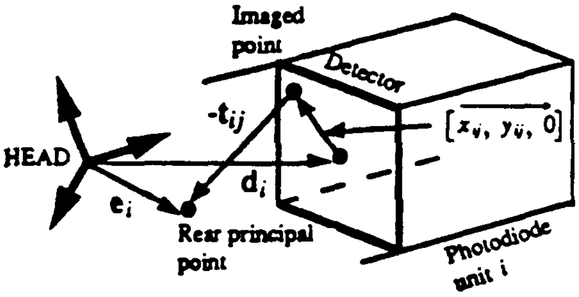


*Figure 11: Expressing $t_{ij}$ through other vectors*

Substituting (2) and (3) into (1) yields the collinearity condition equation $c_{ij}$:

$$c_{ij}: \begin{bmatrix} X_0 - X_j \\ Y_0 - Y_j \\ Z_0 - Z_j \end{bmatrix} + \mathbf{M}\,\mathbf{f}_i = \lambda \, \mathbf{M} \left( \mathbf{d}_i - \mathbf{e}_i + \mathbf{M}_i \begin{bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{bmatrix} \right)$$

### 7.3 System of equations

When a photodiode unit $i$ sees an LED $j$, it generates a $c_{ij}$, which represents three independent equations. If we see $N$ LEDs in all, the total number of unknowns in our system is 6+$N$: 3 for position, 3 for orientation, and $N$ scale factors. The first six are what we are trying to find, but we do not care about the scale factors. We eliminate these by rearranging the $c_{ij}$ equations, then dividing the first and second equations by the third. This leaves two independent equations, of the form

$$G1_{ij}(\mathbf{L}) = 0, \; G2_{ij}(\mathbf{L}) = 0$$

where $\mathbf{L}$ is a vector composed of the six unknowns: position $(X_0, Y_0, Z_0)$ and orientation ($\omega$, $\alpha$, $\kappa$ for matrix $\mathbf{M}$). We generate a linear approximation to these two equations by applying Taylor's theorem:

$$-G1_{ij}(\mathbf{L}) = \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial X_0}\right) dX_0 + \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial Y_0}\right) dY_0 + \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial Z_0}\right) dZ_0 + \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial \omega}\right) d\omega + \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial \alpha}\right) d\alpha + \left(\frac{\partial G1_{ij}(\mathbf{L})}{\partial \kappa}\right) d\kappa$$

and a similar expansion for the linearized $G2$ equation.

Now we have six total unknowns, and every LED that we see generates two independent linear equations. Thus, we need to see at least three LEDs. If we see a total of $N$ LEDs, we can write our system of $N$ linearized $G1$ equations and $N$ linearized $G2$ equations in matrix form:

$$\underset{2N\text{x}1}{-G_0} = \underset{2N\text{x}6}{\partial G} \cdot \underset{6\text{x}1}{D} \qquad (4)$$

where $D = [dX_0, dY_0, dZ_0, d\omega, d\alpha, d\kappa]^T$,
$\partial G$ is the matrix of partial derivatives of the $G1$ and $G2$,
and $-G_0$ contains the values of the $G1$ and $G2$ at a specific L.

### 7.4 Iteration and convergence

Collinearity takes an initial guess of L (the unknowns) and generates correction values (in D) to make a more accurate L, iterating until it converges to a solution. Thus, we need to extract D from equation (4). If $N = 3$, then we can solve for D directly. If $N > 3$, then the system is overdetermined and we approximate D through singular value decomposition [24]. Simulations show that using more than the minimum of 3 LEDs can reduce average error caused by non-systematic error sources. In pseudocode, our main loop is:

```
Generate an initial guess for L
repeat
    Given L, compute $G_0$ and $\partial G$
    Estimate D using singular value decomposition
    $L = L + D$
until magnitude of D is small
return L
```

How do we generate the initial guess of L? Normally we use the last known position and orientation, which should be an excellent guess because we track at rates up to 100 Hz. Collinearity usually converges in 1 or 2 iterations when the guess is close. But in degenerate cases (at system startup, or when we lose tracking because the photodiode units are pointed away from the ceiling), we have no previous L. Collinearity will not converge if the guess is not close enough to the true value; we empirically found that being within 30° and several feet of the true L is a good rule of thumb. So in degenerate cases, we draw initial guesses for L from a precomputed lookup table with 120 entries, trying them sequentially until one converges. We can double-check a result that converges by comparing the set of LEDs used to generate that solution to the theoretical set of LEDs that the photodiode units should see, if the head actually was at the location just computed. When these two sets match, we have a valid solution.

## 8 Performance

A "typical situation" is defined as a user of average height standing erect underneath the ceiling, with at least three photodiode units aimed at the ceiling, moving his head at moderate speeds. All measurement bounds assume that the user remains in tracker range with at least two sensors aimed at the ceiling.

*Update rate:* The update rate ranges between 20–100 Hz. Under typical situations, 50-70 Hz is normal, depending on the height of the user. The wide variation in the .umber of LEDs seen by the sensors causes the variation in update rate. The more LEDs used, the slower the update rate, because LED Manager is the slowest step in the pipeline. If the head remains still and the sensors see a total of $B$ beacons, LED Manager requires 3.33 + 0.782*$B$ ms to run. Rapidly rotating the head increases this time by a factor of about 1.33, since additional time is required to handle the changing working sets of LEDs. Slower head movement rates have correspondingly smaller factors.

*Lag:* Lag varies between 20–60 ms, with 30 ms being normal under typical situations. Lag is measured from the time that LED Manager starts to the time when the Collinearity module provides a computed head location to the graphics engine. Therefore, tracker latency is a function of the number of LEDs seen and the quality of the initial guess provided to the Collinearity module. As $B$ gets smaller, both the LED Manager and Collinearity modules become faster, reducing latency. This mutual dependence on $B$ means that update rate and lag are closely tied: faster update rates correspond with lower latency values.

*Resolution:* When moving the head unit very slowly, we observed a resolution of 2 mm in position and 0.2 degrees in orientation. Measuring accuracy is much harder, and we do not have any firm numbers for that yet. At SIGGRAPH '91, users were able to touch a chair and the four ceiling support poles based solely on the images they saw of models of the chair and the poles in the virtual environment.

## 9 Evaluation

The system provides adequate performance but has several limitations and problems that must be addressed. The most noticeable is the combination of excessive head-born weight and limited head rotation range. Rotation range depends heavily on the user's height and position under the ceiling. A typical maximum pitch range near the center of the ceiling is 45 degrees forward and 45 degrees back. When the user walks near an edge of the ceiling, head rotation range becomes much more restricted. To accommodate the full range of head motion, multiple image sensors must be oriented such that wherever the head is pointed, two or more sensors are able to view LEDs on the ceiling. Given the current focal lengths, simulations show that as many as eight fields of view are required for a respectable rotation range [29]. The weight of each sensor must be significantly reduced to achieve this goal.

To reduce weight, we are trying to replace the current lenses (11 oz. each) with smaller, lighter lenses (2 oz. each). Other approaches are possible. Wang proposed optically multiplexing multiple fields of view onto on a single lateral-effect photodiode [29]. Reduced signal strength, distortions, and view identification ambiguities make this a nontrivial task. It may be easier to design a helmet with integral photodiodes and lenses. Given that each photodiode is about the size of a quarter, the entire surface of a helmet could be studded with sensors.

Beacon switching error has been greatly reduced, but not eliminated. Small observable discontinuities occasionally occur, and while they are not a major disturbance, they are annoying. Calibration techniques are being explored to estimate error sources and compensate for their effects. Photogrammetric techniques like the bundle adjustment method [8] or an alternate scheme suggested by our colleagues [18] may provide the answer.

Infrared light sources in the environment surrounding the tracker, such as sunlight or incandescent light, must be controlled for the system to operate correctly. Specifically, any light source whose wavelengths include 880 nm will be detected by the photodiodes as if it were an LED. For this reason, fluorescent ambient lighting is preferred. Extreme caution is not required, however. Whereas a sensor pointed directly at an infrared light source other than the LEDs will confuse the system, a certain level of indirect infrared background light is tolerable due to the combination of optical filters and the ambient light rejection techniques described in Section 4.

Surprisingly, the bottleneck in the system is the time required to extract data from the photodiode detectors, not the time required to compute the head's location. The i860 processor performs the latter task adequately, and even faster and cheaper processors will be available in the future. But getting accurate photocoordinates from the detectors takes longer than expected, because of the time spent in current scaling and in sampling multiple times per LED. Further experimentation is required to see if we can safely reduce the number of samples. Optimizing the low-level software may improve sampling speed by 20-30%.

The use of Euler angles in the collinearity equations opens the possibility of gimbal lock. The current system avoids this because the head rotation range is too limited to reach gimbal lock positions, but a future version may. If we cannot place the gimbal lock positions out of reach, we can solve for the nine rotation matrix parameters individually, subject to six constraints that keep the matrix special orthogonal, or we may be able to recast the rotations as quaternions.

Since this tracker encourages the user to walk around large spaces, tripping over the supporting cables is a danger. We will investigate the feasibility of a wireless datalink to remove this problem.

Under certain circumstances, the sensors can see large numbers of beacons, such as a total of 30 or more. While using many LEDs usually improves the solution from the Collinearity module, it also slows down the update rate and increases the lag. Further experiments are needed to explore this tradeoff and determine rules of thumb that provide a reasonable balance between resolution and update rate.

Cellular systems using different technologies or configurations could be built to achieve similar scalable work areas. For example, Ascension has announced a cellular magnetic system [4]. Regardless of the technology, any cellular approach creates the problem of beacon switching error or its equivalent. Steps we took to control these errors would apply to other technologies as well: 1) precise positioning and measurement of system components, 2) averaging techniques to reduce random error sources, and 3) calibration routines to compensate for systematic error sources.

## 10 Future work

We intend to continue improving this system. In addition to the tasks listed in Section 9, we would eventually like to expand the ceiling size to around 20' x 20', to provide much greater range of movement, both quantitatively and psychologically. Also, ample room exists to improve the heuristics and optimize the code, increasing the update rate and reducing latency.

But beyond these incremental improvements, we do not expect to pursue this particular technology further. The system is a vehicle for further research and provides room-sized tracking capability today for HMD applications that require it. For example, the UNC Walkthrough team has begun interview-based user studies on what impact large-environment tracking has on the architectural design of a kitchen. In the future, emphasis will be placed on technologies that allow unlimited tracking volumes in unstructured environments. This potential exists in systems that measure only the relative differences in position and orientation as the user moves, integrating these differences over time to recover the user's location. Examples include inertial technologies and Self-Tracker. Since these technologies suffer from drift problems, initial versions may be hybrid systems reliant on the optical tracker for auxiliary information. Thus, the optical tracking system will serve as a testbed for its own successor.

Tracking HMDs will only get harder in the future. The higher resolution displays being developed demand higher resolution trackers. See-through HMDs add additional requirements. In the completely-enclosed HMDs commonly used today, the entire world is virtual, so resolution is much more important than accuracy. But for a see-through HMD, accurate registration of the HMD to the real world is vital. The effects of latency will also become more disturbing in see-through HMDs. Viewing computer-generated objects superimposed upon the real world, where those objects move with significant lag but the real world does not, will not provide a convincing illusion. People can perceive as little as 5 ms of lag [15], and it is unlikely that the combined tracker and graphics engine latency will be below that anytime soon. Therefore, compensation techniques need to be explored [19][24]. If HMDs are to achieve their potential of making a user truly feel immersed inside a virtual world, significant advances in tracking technologies must occur.

## References

[1] Advanced Micro Devices. Am7968/Am7969 TAXIchip Article Reprints, Sunnyvale, CA.

[2] Antonsson, E. K., and R.W. Mann. Automatic 6-D.O.F. kinematic trajectory acquisition and analysis. *J. Dynamic Systems, Measurement, and Control*, 111, (March 1989) pp. 31-39.

[3] Ascension Technology Corporation. The Bird 6D Input Device, Burlington, Vermont, 1989.

[4] Ascension Technology Corporation. A Flock of Birds product description sheet, Burlington, Vermont, April 1991.

[5] Axt, Walter E. Evaluation of a pilot's line-of-sight using ultrasonic measurements and a helmet mounted display. Proceedings IEEE National Aerospace and Electronics Conf. (Dayton, OH, May 18-22, 1987) pp. 921-927.

[6] Azuma, Ronald, and Mark Ward. Space-resection by collinearity: mathematics behind the optical ceiling head-tracker. UNC Chapel Hill Dept. of Computer Science technical report TR 91-048, Nov. 1991.

[7] Bishop, Gary and Henry Fuchs. The self-tracker: A smart optical sensor on silicon. *Proceedings of the 1984 MIT Conference on Advanced Research on VLSI* (Dedham, MA: Artech House, Jan 1984) pp. 65-73.

[8] Burnside, C. D. *Mapping from Aerial Photographs*. Granada Publishing Limited, G. Britain, 1979, pp. 248-258.

[9] Chung, Jim, Mark Harris, Fred Brooks, et al. Exploring Virtual Worlds with Head-Mounted Displays. SPIE Proceedings vol. 1083 Non-Holographic True 3-Dimensional Display Technologies (Los Angeles, CA, Jan 15-20, 1989).

[10] Church, Earl. Revised geometry of the aerial photograph. *Bulletins on Aerial Photogrammetry*, No. 15, Syracuse University, 1945.

[11] Cook, Anthony. The helmet-mounted visual system in flight simulation. Proceedings Flight simulation: Recent developments in technology and use. (Royal Aeronautical Society, London, England, Apr. 12-13, 1988) pp. 214-232.

[12] Fake Space Labs, Binocular Omni-Orientation Monitor (BOOM), Menlo Park, CA.

[13] Ferrin, Frank J. Survey of helmet tracking technologies. SPIE Vol. 1456 Large-Screen Projection, Avionic, and Helmet-Mounted Displays (1991) pp. 86-94.

[14] Fuchs, Henry, John Poulton, John Eyles, et. al. Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories. Proceedings of SIGGRAPH '89 (Boston, MA, July 31-Aug 4, 1989). In *Computer Graphics* 23, 3 (July 1989) pp. 79-88.

[15] Furness, Tom, and Gary Bishop. Personal communication.

[16] Hamamatsu. Hamamatsu Photonics, Hamamatsu City, Japan, 1985.

[17] Hardyman, G. M. and M. H. Smith. Helmet mounted display applications for enhanced pilot awareness. Proceedings of AIAA Flight Simulation Technologies Conference (Boston, MA, Aug. 14-16, 1989) pp. 221-225.

[18] Hughes, John F., and Al Barr. Personal communication.

[19] 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.

[20] Logitech, Inc. Logitech 3-D Mouse news release. July 30, 1991.

[21] POLHEMUS 3SPACE User's Manual, OPM3016-004B, Colchester, Vermont, 1987.

[22] Press, William, Brian Flannery, Saul Teukolsky, William Vetterling. *Numerical Recipes in C*. Cambridge University Press, USA, 1988.

[23] SELCOM. SELSPOT II HARDWARE and MULTILab Software, Southfield, Michigan, 1988.

[24] Smith Jr., B. R. Digital head tracking and position prediction for helmet mounted visual display systems. Proceedings of AIAA 22nd Aerospace Sciences Meeting, (Reno, NV, Jan. 9-12, 1984).

[25] So, Richard H., and Michael J. Griffin. Effects of time delays on head tracking performance and the benefits of lag compensation by image deflection. Proceedings of AIAA Flight Simulation Technologies Conference (New Orleans, LA, Aug. 12-14, 1991) pp. 124-130.

[26] Sorensen, Brett, Max Donath, Guo-Ben Yang, and Roland Starr. The Minnesota scanner: a prototype sensor for three-dimensional tracking of moving body segments. *IEEE Transactions on Robotics and Automation*, 5, 4, (August 1989), pp. 499–509.

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

[28] Wang, Jih-Fang, Vernon Chi, and Henry Fuchs. A real-time 6D optical tracker for head-mounted display systems. Proceedings of 1990 Symposium on Interactive 3D Graphics (Snowbird, Utah, 1990). In *Computer Graphics* 24, 2 (March 1990) pp. 205-215.

[29] Wang, Jih-Fang, Ronald Azuma, Gary Bishop, Vernon Chi, John Eyles, Henry Fuchs. Tracking a head-mounted display in a room-sized environment with head-mounted cameras. SPIE Proceedings Vol. 1290 Helmet-Mounted Displays II (Orlando, FL, Apr 19-20 1990) pp. 47-57.

[30] Welch, Brian, Ron Kruk, Jean Baribeau, et al. Flight Simulator: Wide-Field-Of-View Helmet-Mounted Infinity Display System, Air Force Human Resources Laboratory technical report AFHRL-TR-85-59, May 1986, pp. 48-60.

[31] Wolf, Paul. *Elements of Photogrammetry, With Air Photo Interpretation and Remote Sensing*, 2nd ed., McGraw-Hill, New York, 1983.

[32] Woltring, Herman. Single- and Dual-Axis Lateral Photodetectors of Rectangular Shape. *IEEE Trans. on Electron Devices*, (August 1975) pp. 581–590.

## A Guide to Using the Demo:

# Interactive Building Walkthrough Using the Optical Tracker

UNIVERSITY OF NORTH
CAROLINA AT
CHAPEL HILL
Department of
Computer Science

Walkthrough is a system to help architects and their clients explore a building design prior to its construction, correcting problems on the computer instead of in concrete. To sustain the illusion of being inside a virtual building, we must make it possible to move through the building in a natural manner, with as few encumbrances as possible. The head-mounted display ana optical tracker allow a large working volume where the user can walk naturally in the virtual building. Goals of optical tracker research include using see-through optics in the head-mounted display and tracking in unstructured environments, such as an outdoor building site.

## Instructions

Use these guidelines to help you experience the Optical Tracker demonstration. The goal of the demonstration is to allow you to fully experience a virtual world application in the limited time available.

1) Sit in the special chair that is bolted to the floor. A UNC assistant will place the head-mounted display on your head and will remove it after the demonstration. The headgear is very heavy (10 lbs). You may want to hold the VPL EyePhone portion to steady it during the demonstration. Please do not touch the rest of the head-mounted display; it is a prototype and is liable to damage.
2) Be careful to notice the limits of motion. The tracker does not track as well near the edges of the ceiling, because one or more sensors may be off the ceiling. The ceiling is about the size of the models, so when you are near the walls of the model you are near the edge of the ceiling.
3) Walk!
4) At the end of the 3-minute demo, walk back to the chair and try to sit back down in it. The chair exists in the computer model of the simulated environment. Use the computer-generated images you see in the head-mounted display to guide you.

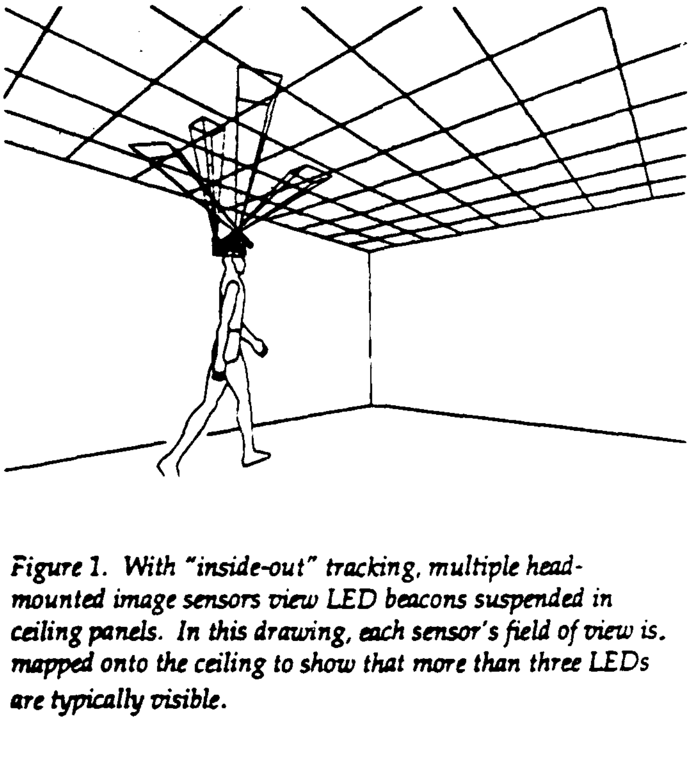


*Figure 1. With "inside-out" tracking, multiple head-mounted image sensors view LED beacons suspended in ceiling panels. In this drawing, each sensor's field of view is. mapped onto the ceiling to show that more than three LEDs are typically visible.*

## The Tracker

An optoelectronic tracking system that adopts an *inside-out* paradigm introduced by Gary Bishop and Henry Fuchs in 1984 is used in the demonstration. The current prototype features 2-x 2-foot ceiling panels in a 10-x 12-foot room with 32 individually addressable light emitting diodes (LEDs) mounted in each panel. The LEDs serve as a global position refercnce for a head-mounted display that is augmented with four, upward looking image sensors. Each sensor resembles a small camera in that its lens focuses infrared light onto a lateral-effect photodiode's surface. The photodiode, in turn, reports the photocoordinates of an LED that has been momentarily turned on. The photocoordinates associated with three or more LEDs, along with their absolute position in the room, allow the position and orientation of the head-mounted display to be computed. The sample rate of the optical tracker is dependent on the number of LEDs imaged and ranges from 35 to 100 Hz. A 30 to 50 ms lag in the calculated measurements is caused by a two stage pipeline in the architecture.

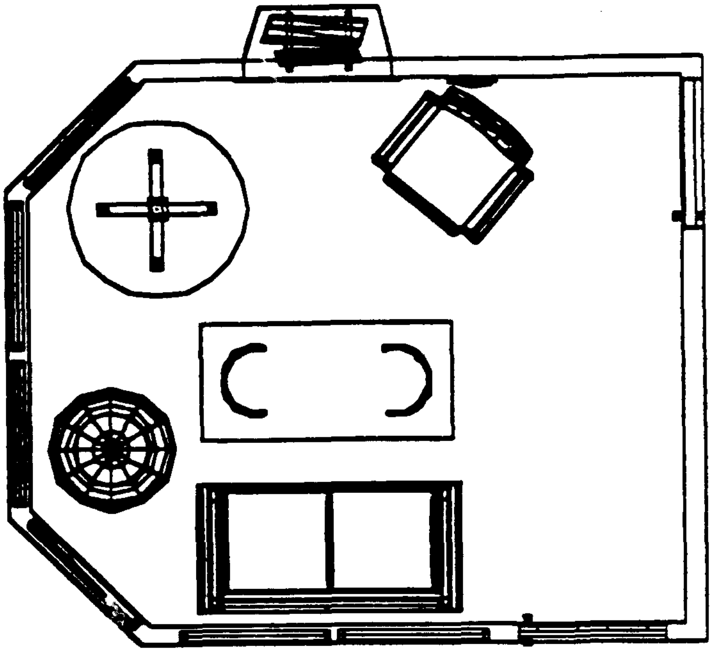


*Figure 2. Floor plan of the living room of a hypothetical apartment model.*

The advantage of this approach lies in its cellular nature. Typically, optical tracking is done with fixed cameras and moving LEDs. That is, the cameras are *outside* looking "in." A fundamental limitation exists with this approach: the working volume is limited to the cameras' field of view. Furthermore, to achieve high resolution, long focal lengths are required and a reduction in working volume follows. Reversing the situation, i.e., moving the cameras and fixing multiple LEDs, solves this dilemma. Long focal lengths can be used to achieve high resolution and, as the cameras move, the set of LEDs in a camera's field of view is constantly changing. This unique concept, diagrammed in Figure 1, is the first step in a long range research effort. Future goals will involve reducing the number of beacons required by utilizing sensors that are capable of measuring incremental head motion. Both optical and inertial technologies are being investigated.

The ultimate goal in user interface should be to make the interface as transparent as possible to the user. Interaction with the model should be as easy and intuitive as interacting with the real object. Thus, the devices should encourage interaction with the model in a natural manner. In this the optical tracker hopes to excel.

## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs
Walkthrough Software: John Alspaugh, Amitabh Varshney, Yulan Wang, Xialin Yuan
Tracker Hardware: Vern Chi, John Eyles, Jack Kite, Mark Ward
Tracker Software: Ron Azuma, Carney Clegg, Stefan Gottschalk, Phil Jacobsen
Additional help: John Airey, Brad Bennett, Gary Bishop, Randy Brown, Andrew Davidson, Curtis Hill, Rich Holloway, John Hughes, Jannick Rolland, John Thomas, and the Pixel-Planes 5 team.
Sponsors: Defense Advanced Research Projects Agency, National Institutes of Health, National Science Foundation, Office of Naval Research

# A Guide to Using the Demo:

# 3dm: A Two-Person Modeling System

UNIVERSITY OF NORTH CAROLINA AT
CHAPEL HILL
**Department of Computer Science**

**3dm** is a program that allows the creation of a variety of virtual geometric objects. You can fly around in this virtual world in which the other user, the designer, is creating and modifying objects according to your verbal instructions. A variety of interesting things can be created rapidly with **3dm**, especially things composed of boxes, spheres, and cylinders. You can ask the modeler to:

- change the color of an object. For example, "make that sofa green."
- copy and paste objects to make multiple copies. For example, "make a few copies of that tree.'
- make an object bigger or smaller or reshape part of an object. For example, "make the tree bigger" or "make that branch point down more."
- create objects like the following: the Old Well, a tree, a potted plant, a rock, mountains, skyscrapers, a water tower, a simple house, a castle, a sofa or chair, a table, a fish, a jet or rocket.

## Instructions

Use these guidelines to help you experience the 3D modeling system. The goal of the demonstration is to allow you to fully experience a virtual world scenario in the limited time available.

1) Let the UNC assistant help put the VPL EyePhone on your head. This is done by separating the earphones on the EyePhone while you slip on the headgear. Make sure that the EyePhone fits snugly.

2) You will be holding a 3D mouse with two buttons mounted on it. In the virtual world this mouse will appear as a cursor in the shape of a small airplane. For best results, hold your hand at least 6 inches from the EyePhone. Holding your hand too close to the EyePhone will interfere with the Polhemus tracking sensor and your hand or the airplane will shake.

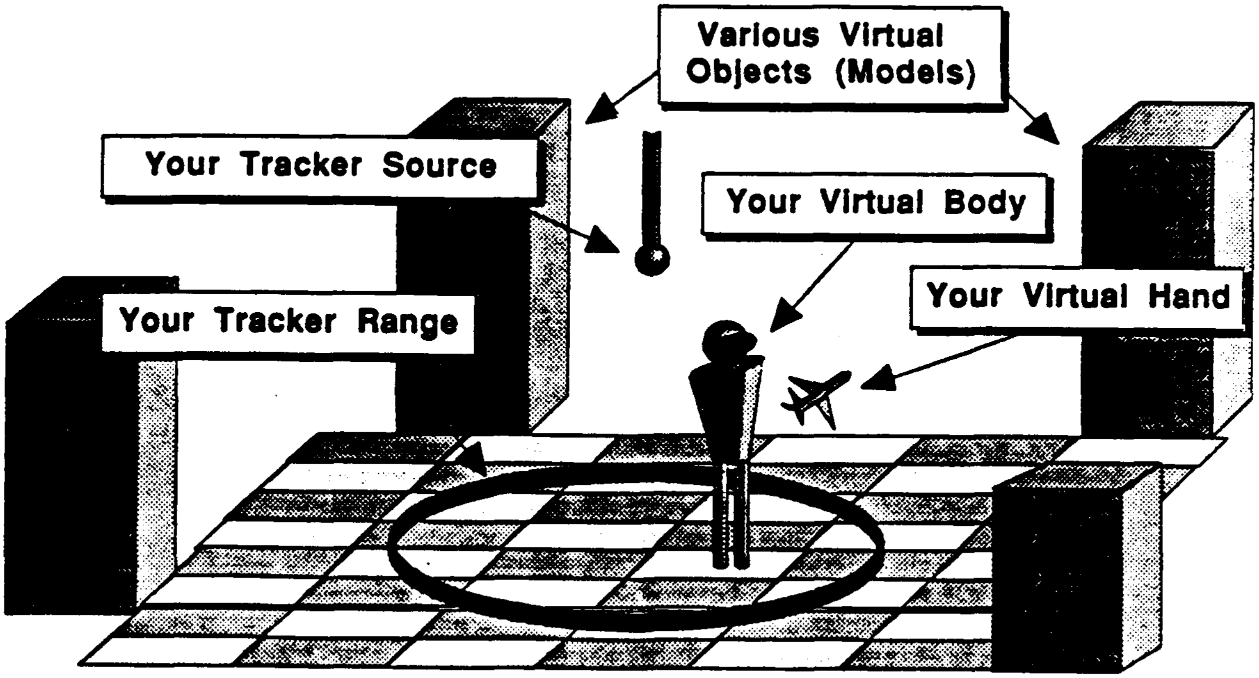
3) You will notice that a yellow ring surrounds you on the floor. This represents the range of your Polhemus tracker—if you walk beyond it (or close to its edge), Polhemus can no longer track your position accurately and the image that you see of your virtual world will shake violently. Walk back within the ring to restore normal tracking. For best tracking response, stand near the Polhemus tracking source, which hangs overhead.

4) The modeler in the virtual world is surrounded by a red ring on the floor, which represents his or her tracking range. The modeler's head and body will appear in red. You will be able to see the modeler's cursor, which will take on different shapes and sizes. The modeler also has a toolbox with tools that allow him or her to create objects. You will not be able to choose tools from the toolbox.

5) To fly forward (the direction the airplane is pointing), hold down the left mouse button. To fly backward, hold down the right mouse button. Your speed will slowly increase the longer you hold down either mouse button. To stop flying, release the mouse button.

6) Because you will be flying in the direction your hand is pointing and not in the direction

## View of the 3dm Virtual World



*(Note: The Modeler's virtual body, hand, range, source, and toolbox are not shown. The various virtual models will not necessarily appear as those in this diagram.)*

you are looking, you may look anywhere while flying without affecting your motion. Also, remember that you fly in the direction your airplane points, so for most accurate flying it helps to "sight down" the airplane (that is, look at it from directly behind its tail) to see exactly where it is pointing.

7) You will see several objects that have been created in the virtual world. You may fly around, through, and into these objects, or anywhere in the virtual world. You can fly up or down and will not fall. There is no gravity in this virtual world. If you fly too far from the model, you may get lost. Scan the virtual world by looking around to find the model so that you can fly back to it. Remember to look up and down!

8) When the virtual world freezes and the words "Game Over" appear, please wait for the UNC attendant to help you take off the EyePhone.

## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs

Authors: Jeff Butterworth, Andrew Davidson, Stephen Hench, T. Marc Olano

Thanks to: Warren Robinett, for being 3dm's client for Comp 145, "Software Engineering"
Richard Holloway, for vlib, trackerlib, and adlib

Sponsors: Defense Advanced Research Projects Agency, National Science Foundation,
Office of Naval Research

## A Guide to Using the Demo:

# Flying Through Molecules

UNIVERSITY OF NORTH
CAROLINA AT
CHAPEL HILL
Department of
Computer Science

The intent of this application is to give you an "atom's perspective" of an assortment of molecules, while demonstrating a method of navigating through a large space in a virtual world. You will use a head-mounted display and a hand-held input device to fly through a virtual world containing the following giant molecules:

- *DHFR (dihydrofolate reductase) and methotrexate*– DHFR is a protein molecule and methotrexate is a drug molecule used in cancer treatment, which is known to dock with DHFR. Both are represented with CPK (sphere-only) models.
- *crambin*– a protein molecule, represented here by three different models: CPK (spheres represent individual atoms colored according to type), stick-only (sticks represent the bonds between the atoms), and ribbon (the amino acid chain that makes up the backbone of the model is represented as a ribbon).
- *clathrate*– a highly energetic compound which is represented as a ball-and-stick model.

## Instructions

Use the following guidelines to experience this virtual worlds application. The goal of the demonstration is to allow you to fully experience a virtual world scenario in the limited time available.

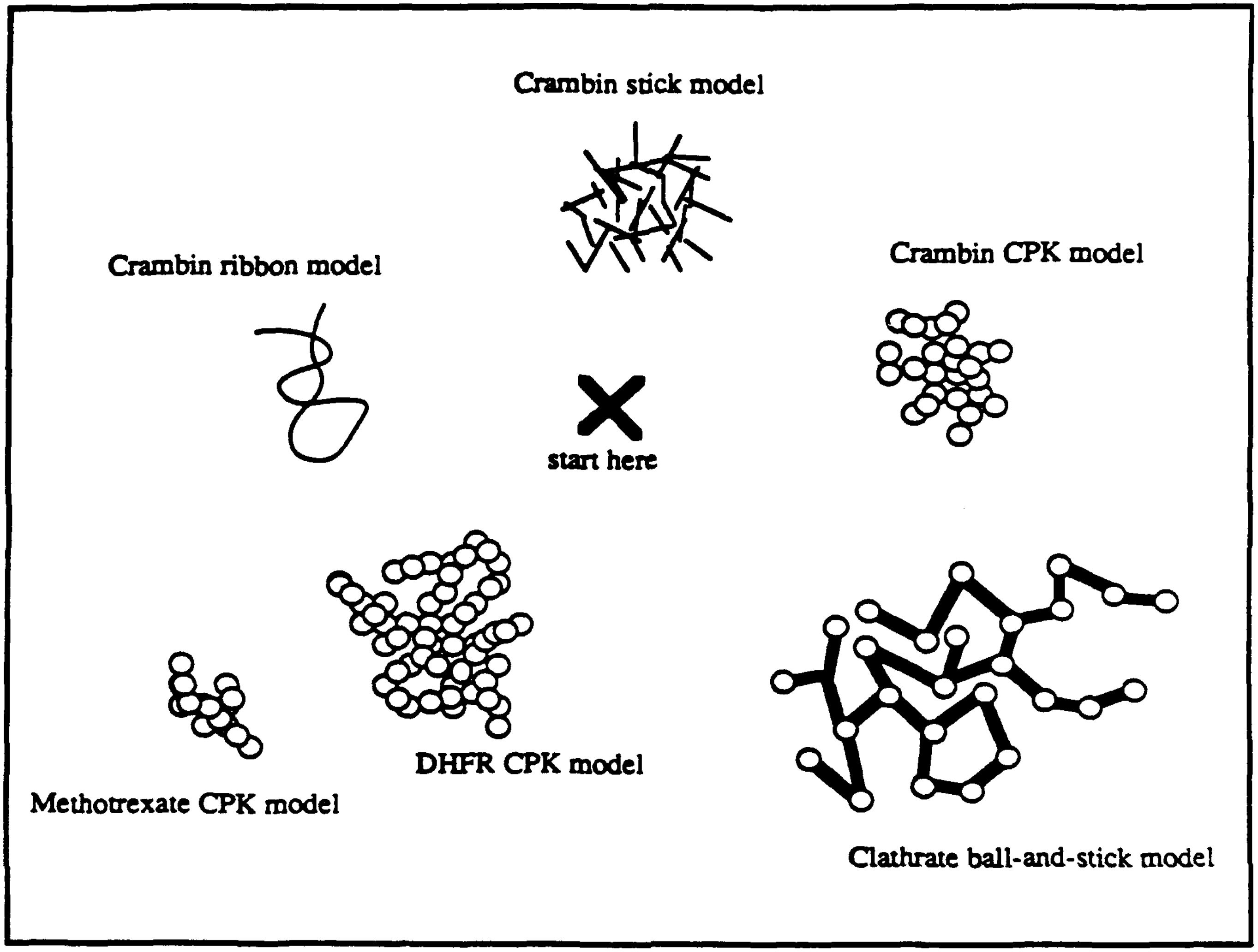
1) Let the UNC assistant help put the VPL EyePhone on your head. This is done by separating the earphones on the EyePhone while you slip on the headgear. Make sure that the EyePhone fits snugly.

2) To "fly" through the molecules, grab the 3D mouse and point your hand in the direction you want to go. Press the left button to go forward; the right button to go in reverse.

3) When you press a button to fly in either direction, you will hear a "rocket" sound. This indicates that you are flying.

4) Your speed is proportional to the amount of distance you put between your hand and your body; i.e., to fly faster, you hold the mouse further away from you and press the button. You can check your speed by looking at the color of the hand icon, which is in the shape of an arrow. At minimum speed, the arrow is blue; at maximum speed, it is red. The color blends smoothly for all speeds in between. When you are not pressing any buttons, the arrow will be white. **Be sure to hold your hand out in front of you so that you can see the direction and color of the arrow.**

## Bird's Eye View of Molecule Flythrough Environment



## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs

Authors: Richard Holloway, Warren Robinett

Input device: Jack Kite, John Hughes

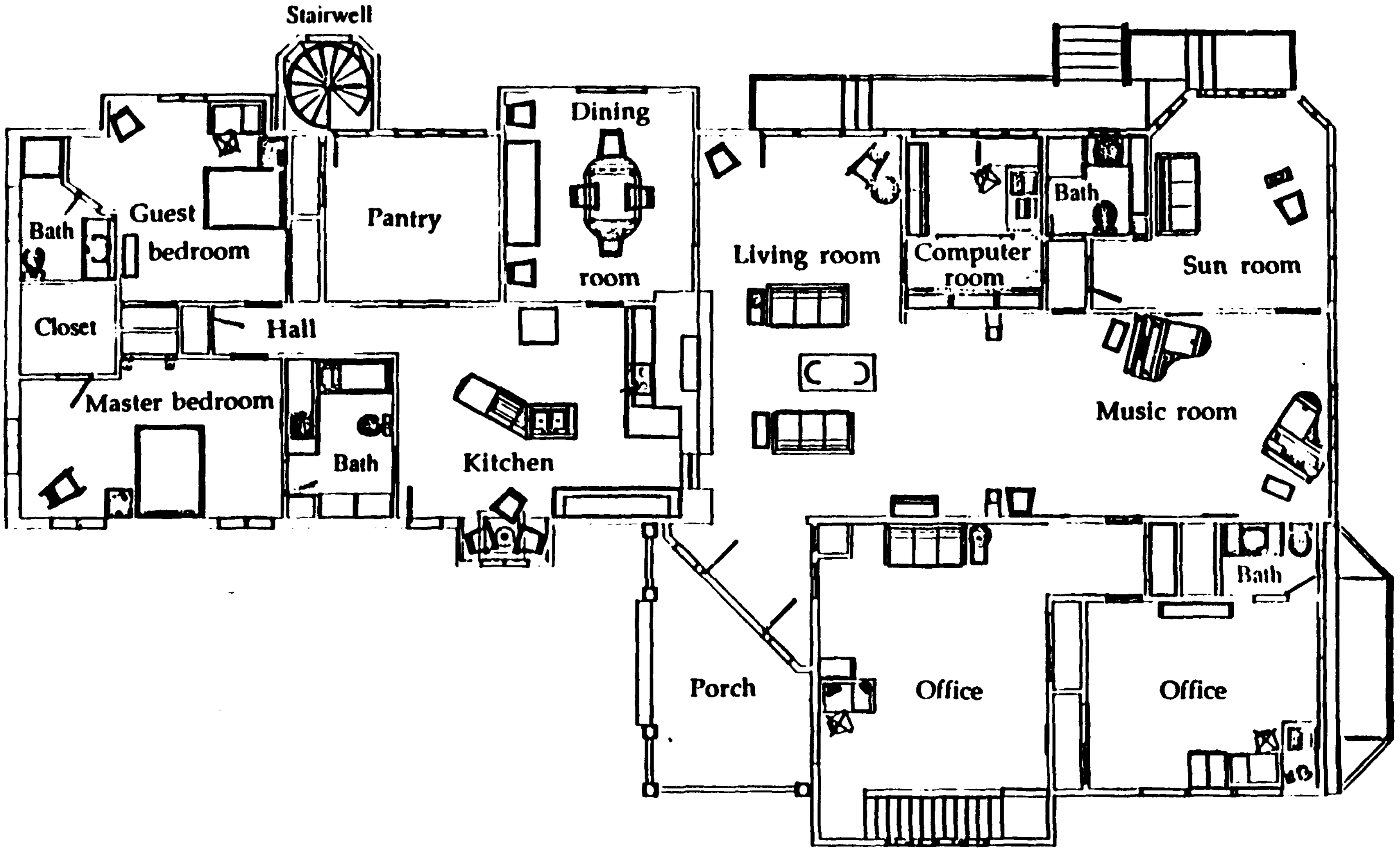
Sound library: Xialin Yuan, Andrew Davidson

Flying speed paradigm suggested by Mark Levoy

Molecular data and models:

- DHRF & methotrexate: (3DFR & 4DFR from the Brookhaven Data Bank), Lee Kuyper, Burroughs-Wellcome
- clathrate: Arnie Nielsen, Naval Weapons Center; Robert Schmitt, Stanford Research Institute; Richard Gilardi, Naval Research Lab
- crambin: (1CRN from the Brookhaven Data Bank), Wayne Hendrickson, Martha Teeter, Joseph Kraut

Sponsors: National Institutes of Health, Defense Advanced Research Projects Agency, National Science Foundation, Office of Naval Research



Floor plan of a proposed renovation of the Brooks house

# A Guide to Using the Demo:

# Interactive Building Walkthrough Using A Steerable Treadmill

UNIVERSITY OF NORTH
CAROLINA AT
CHAPEL HILL
Department of
Computer Science

Walkthrough is a system designed to help architects and their clients explore a building design prior to its construction, correcting problems on the computer instead of in concrete. Using a treadmill and a head-mounted display, the user can navigate through a three-dimensional virtual building and evaluate its design. A treadmill is an appropriate vehicle to use for this task since it both immerses the user in the virtual environment and allows him to explore it in a natural manner–by walking.

## Instructions

Use these guidelines to help you experience the Interactive Building Walkthrough demonstration. The goal of the demonstration is to allow you to fully experience a virtual world application in the limited time available.

1) Let the UNC assistant help put the VPL EyePhone on your head. This is done by separating the earphones while you slip on the headgear. Make sure that the EyePhone fits snugly.

2) Movement through the model is done by walking on the treadmill. Leaning forward slightly and pushing against the handlebars will make walking easier. Your speed is controlled by the length of your stride and how quickly you walk.

3) To turn, turn the handlebars as you would on a bicycle. As on a bicycle, if you turn the treadmill's handlebars sharply, you will turn sharply in the virtual world of the building.

4) Lights in the model can be turned on and off by the UNC assistant. If you want to change the lights in a room, tell the assistant which ones to change.

## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs

Authors: John Alspaugh, Amitabh Varshney, Yulan Wang, Xialin Yuan

Additional help: John Airey, Gary Bishop, Randy Brown, Andrew Davidson, Curtis Hill, Rich Holloway, Harry Marples, Michael Zaretsky, and the Pixel-Planes 5 team

Sponsors: Defense Advanced Research Projects Agency, National Institutes of Health, National Science Foundation, Office of Naval Research

SIGGRAPH '91 August 1991

# A Guide to Using the Demo:

# Radiation Therapy Treatment Planning

UNIVERSITY OF NORTH CAROLINA AT
CHAPEL HILL
Department of
Computer Science

The basic goal of radiation treatment of a cancerous tumor is to irradiate the tumor with enough dosage to kill the malignant cells without further hurting the patient. This demonstration represents ongoing research at UNC using state-of-the-art 3D displays to design radiation therapy treatment plans.

Conventional radiotherapy treatment planning is based on a handful of 2D X-ray images showing prospective "beam's-eye" views of the patient's anatomy. Dose fields for those prospective beam configurations are completed only for isolated 2D slices through the patient's body. The result is suboptimal conformation between the tumor volume and the proposed dose distribution, both of which have complex 3D structures but are studied only in 2D.

Current state-of-the-art 3D treatment planning takes advantage of today's display and computational power to enable the radiotherapist to study the patient in three dimensions. Interactive beam targeting using a computerized model of the patient (as opposed to using the real patient, as in 2D planning) permits the radiotherapist to explore a continuum of beam configurations. 3D dose grids are now computed and registered with the patient anatomy model, providing better comparisons between tumor volume and dose distribution.

Head-mounted displays may be useful for targeting treatment beams since they allow more natural steering and navigation through the patient's anatomy than that allowed by fixed workstation screens.

## Instructions

This demonstration is designed to give you a feel for how a head-mounted display might be used for the task of targeting radiation treatment beams. Use the following guidelines to experience this virtual worlds application. The goal of the demonstration is to allow you to fully experience a virtual world scenario in the limited time available.

1) Let the UNC assistant help put the VPL EyePhone on your head. This is done by pulling the headphones down over your ears while you slip on the headgear. Make sure that the EyePhone fits snugly.

2) You are presented with a virtual patient, a polygonal model of a human figure with a portion of the surface removed to reveal yellow tumors located on the patient's kidneys. Also visible are the patient's liver, spleen, renal artery, spine and ribs.

3) Next to the patient's head are three treatment beams. Your task is to position these three beams so that they irradiate the tumors while missing as much of the sensitive healthy tissue as possible. To manipulate a beam, place the red tip of the hand cursor's index finger into a beam. A wireframe bounding box appears around the beam, indicating that the beam has been selected. A low-frequency hum is also audible. To grab the beam, press either mouse

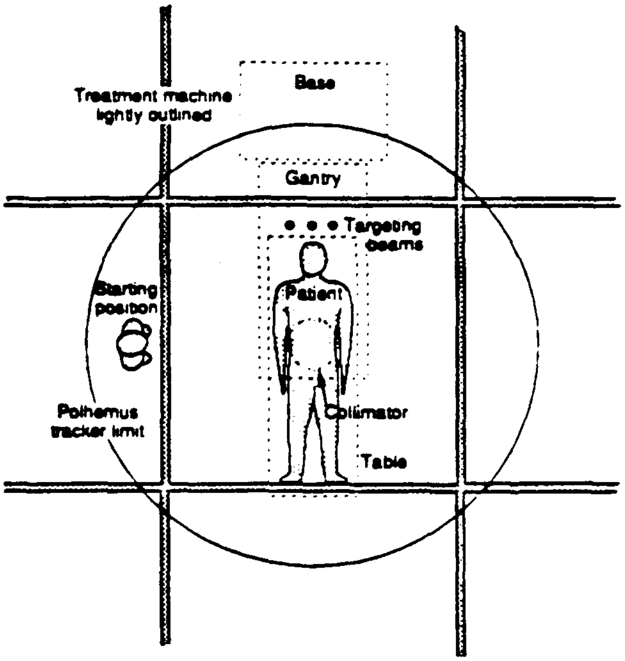
button while the beam is selected. As long as you hold the button down (i.e. your hand is "holding" the beam), the beam is controlled by the mouse. When you release the mouse button, the beam stays in its current position.

4) Grab each beam and position it so that it passes through the tumor, but hits as little of the other anatomical structures as possible. The multiple beams allow you to attack a particular tumor from different directions, so that the tumor is multiply irradiated while healthy tissue is hit by just one beam.

5) Because of the limitations of the treatment machine, which is ultimately used to deliver the beams you are targeting, not all beam directions are valid. When your beam is in an invalid configuration it will be colored cyan, whereas a valid beam configuration is colored red. You have 3 1/2 minutes to target the three beams.

6) When you hear the sound of the harp glissando, your time is up. You then see your treatment plan being administered to the patient. The treatment machine appears. It and the patient are scaled down by 50 percent to give you a better view of the whole setup. Step back and watch as the table and gantry move to deliver the treatment plan you have specified.

**Bird's Eye View of Virtual World**



## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs

Authors: James C. Chung, Bradley A. Crittenden, Suresh Balu, Terry Yoo

Data Courtesy of: Dr. Ron Kikinis; Dr. Ferenc A. Jolesz; Steven Seltzer, and Stuart Silverman,
Brigham & Women's Hospital and Harvard Medical School, Boston,
Bill Lorensen; Harvey Clin, and Jon Zarge, GE Corporate R&D,
VPL Research, Inc.

Sponsors: National Institutes of Health, Defense Advanced Research Projects Agency,
National Science Foundation, Office of Naval Research

# A Guide to Using the Demo:

# Mountain Bike

UNIVERSITY OF NORTH
CAROLINA AT
CHAPEL HILL
Department of
Computer Science

The mountain bike virtual worlds application was originally designed to be an enhanced version of a stationary bicycle, providing both interactive entertainment and exercise for the user. In this scaled-down version, you can "ride" a stationary bicycle over terrain in a virtual world. The bicycle is equipped with pedal force feedback, which makes pedalling uphill more difficult than pedalling downhill, just as in the real world.

## Instructions

Use the following guidelines to experience this virtual worlds application. The goal of the demonstration is to allow you to fully experience a virtual world scenario in the limited time available.

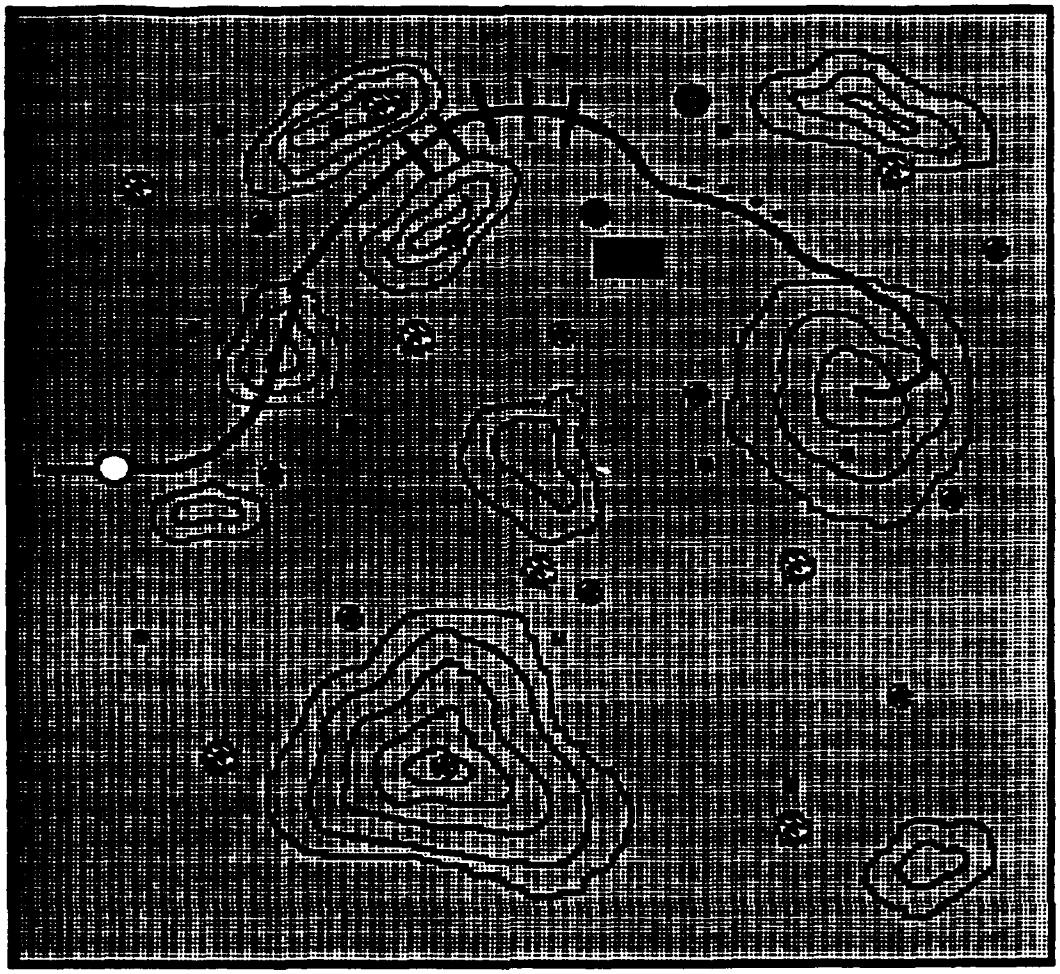
1) Seat yourself on the bicycle.
2) Let the UNC assistant help put the VPL EyePhone on your head. This is done by separating the earphones on the EyePhone while you slip on the headgear. Make sure that the EyePhone fits snugly.
3) When the demonstration begins, you will be overlooking a textured terrain. There is a road that designates a path through the world. If you ride along it, you will be able to see most of the signs, buildings and other interesting objects that are part of this virtual world. You do not have to stay on the road but, since the demonstration is only three minutes long, you will see most of the attractions if you ride on the road.
4) When your three minutes are up, or when you have finished your ride, the words "Game Over" will appear. Wait for the UNC attendant to help you take off the EyePhone.
5) Most importantly, enjoy your ride and happy exercising.

## Hardware details

This application uses a VPL EyePhone for its display device and a Polhemus 3SPACE TRACKER for tracking the location and orientation of the user's head. A Sun-4 workstation is used to interface this equipment to the graphics engine—Pixel-Planes 5—a custom-made, massively parallel imaging machine developed at UNC–Chapel Hill.

The bicycle has sensors mounted on its back wheel and front fork. These sensors measure the user's distance and direction changes. The number of rear wheel revolutions is used to measure distance and speed, and the turning of the handlebars determines the bicycle's direction.

## Mountain Bike Terrain



The bicycle has an eddy-current resistance motor attached to the back wheel; a CompuTrainer, Model CAT 6000 by Racermate.

When the rider is pedalling uphill, pedal resistance increases according to the steepness of the grade. It is not possible to coast downhill in this scenario because the motor that is attached to the bicycle can only affect resistance to the wheel's turning and cannot itself help propel the wheel.

## Credits

Authors: Erik Erikson, Ryutarou Ohbuchi, Andrei State, Russell Taylor.

This application began as a class project by Ryutarou Ohbuchi in Comp 290 (now 239), "Exploring Virtual Worlds" (taught by Henry Fuchs).

Sponsors: Defense Advanced Research Projects Agency, Office of Naval Research

# A Guide to Using the Demo:

# Molecule Museum

UNIVERSITY OF NORTH CAROLINA AT
CHAPEL HILL
**Department of Computer Science**

One of the wonderful things about displaying molecules is that there is no single correct way to do it. Therefore, many different representations are possible, and chemists use different ones depending on the situation.

This application lets you stroll around several different representations of a protein molecule, stick your head inside them, and even pick them up and spin them around. The molecule is called crambin, and is represented here by three different models:

- CPK: spheres represent individual atoms colored according to type
- stick-only: sticks represent the bonds between the atoms
- ribbon: the amino acid chain that makes up the backbone of the model is represented as a ribbon

## Instructions

Use the following guidelines to experience this virtual worlds application. The goal of the demonstration is to allow you to fully experience a virtual world scenario in the limited time available.

1) Let the UNC assistant help put the VPL EyePhone on your head. This is done by separating the earphones on the EyePhone while you slip on the headgear. Make sure that the EyePhone fits snugly.

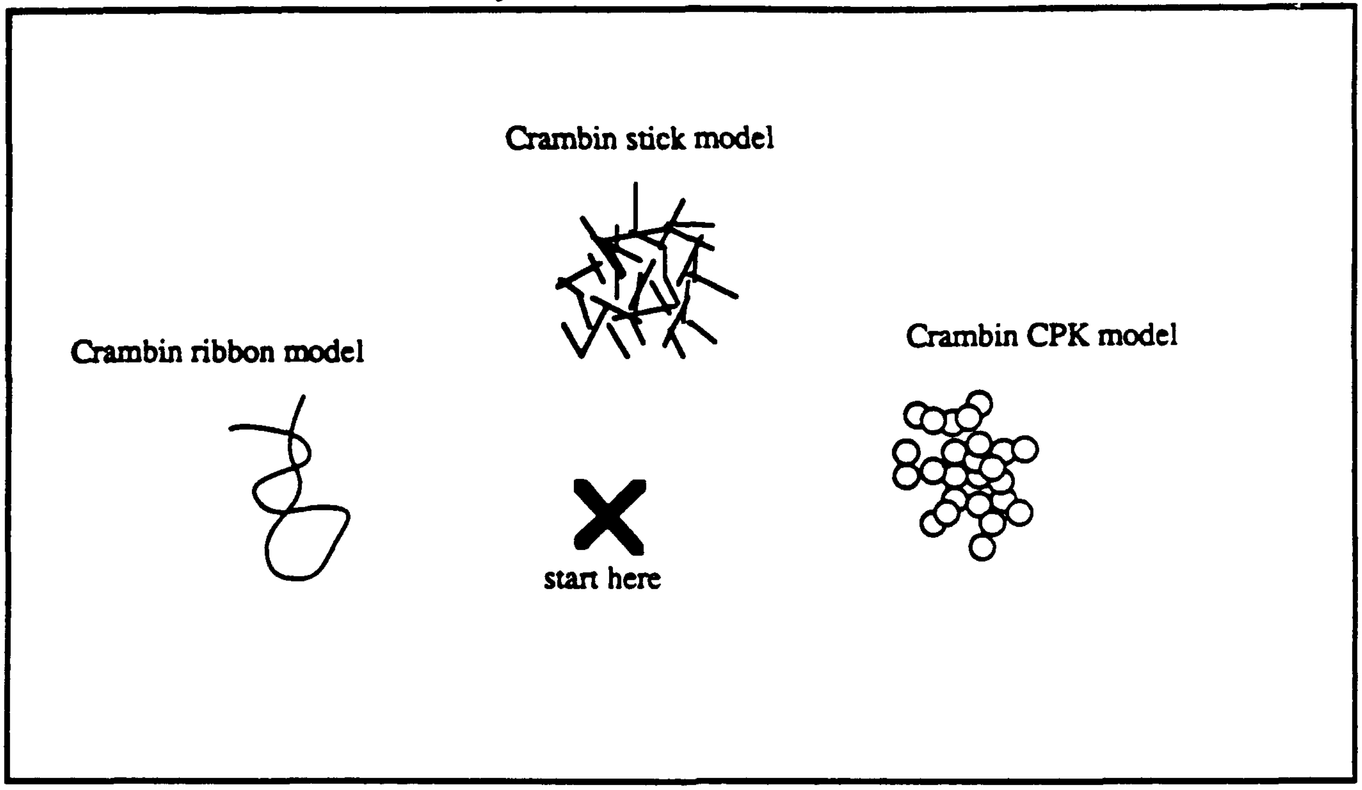
2) **Viewing:** To get a better look at a model, just walk up to it. Putting your head inside the model is strongly encouraged (where else can you put your head inside a molecule?).

3) **Grabbing:** To grab a model, put your hand (while holding the 3D input device) inside the model and press either button and hold the button down. When your hand is inside the model, the model's "bounding box" will be drawn around it to highlight it; in addition, you should hear a low hum to indicate that the model can be grabbed. When you successfully grab a model, you should hear a cartoonish "floop" sound (only at the instant you grab–it doesn't continue like the hum does). For as long as you hold either button down, the model will stay attached to your hand, and will follow your hand's position and orientation.

4) **Releasing:** To release the model, just release the buttons.

5) You may want to try superimposing one or more models on each other to see how the different representations "line up." Do this by grabbing one and putting it in the same spot as the other–you can put all three on top of each other if you wish. When two or more models are piled up together in this manner, you can press the button to grab whichever model is closest to the center of your virtual hand.

## Bird's Eye View of Molecule Museum



## Credits

Principal Investigators: Frederick P. Brooks, Jr., Henry Fuchs

Authors: Richard Holloway, Warren Robinett

Input device: Jack Kite, John Hughes

Sound library: Xialin Yuan, Andrew Davidson

Molecular data and models:

- Crambin: (1CRN from the Brookhaven Data Bank), Wayne Hendrickson, Martha Teeter, Joseph Kraut

Sponsors: National Institutes of Health, Defense Advanced Research Projects Agency,
National Science Foundation, Office of Naval Research