Incorporating the Viewer's Point-Of-Regard (POR) in Gaze-Contingent Virtual Environments*

Andrew T. Duchowski†
andrewd@cs.tamu.edu
Department of Computer Science
Texas A&M University
College Station, TX, 77843-3112

ABSTRACT

Awareness of the viewer's gaze position in a virtual environment can lead to significant savings in scene processing if fine detail information is presented "just in time" only at locations corresponding to the participant's gaze, i.e., in a *gaze-contingent* manner. This paper describes the evolution of a gaze-contingent video display system, "`gcv`". `Gcv` is a multithreaded, real-time program which displays digital video and simultaneously tracks a subject's eye movements. Treating the eye tracker as an ordinary positional sensor, `gcv`'s architecture shares many similarities with contemporary virtual environment system designs. Performance of the present system is evaluated in terms of (1) eye tracker sampling latency and video transfer rates, and (2) measured eye tracker accuracy and slippage. The programming strategies developed for incorporating the viewer's point-of-regard are independent of proprietary eye tracking equipment and are applicable to general gaze-contingent virtual environment designs.

Keywords: Gaze-Contingent, Virtual Environments, Eye Tracking.

1 INTRODUCTION

This paper describes a gaze-contingent video display system, `gcv`, developed for the purpose of recording eye movement data during simultaneous real-time video viewing. `Gcv` is a multithreaded, real-time program which displays digital video and simultaneously tracks a subject's eye movements. Treating the eye tracker as an ordinary positional sensor, `gcv`'s architecture shares many similarities with contemporary virtual environment system designs. The paper presents problems encountered and lessons learned during system development.

Hardware and software configurations are described in §2 and §3, respectively. Calibration procedures, designed to be independent of proprietary eye tracking equipment methods, are described in §4. Eye tracker to image coordinate space mapping of the measured Point Of Regard (POR) is given in §4.2. In §5, performance of the present system is evaluated in terms of (1) eye tracker sampling latency and video transfer rates, and (2) measured

* This research was supported in part by the National Science Foundation, under Infrastructure Grant CDA-9115123 and CISE Research Instrumentation Grant CDA-9422123, and by the Texas Advanced Technology Program under Grant 999903-124.

† As of 01/98 the author can be reached at *andrewd@cs.clemson.edu*, Department of Computer Science, 451 Edwards Hall, Clemson University, Clemson, SC 29634-1906.

eye tracker accuracy and slippage.

2 HARDWARE

The main hardware components of the eye movement recording system include an ISCAN eye tracker, an SGI Onyx® RealityEngine2™ host computer with two MIPS® R4400™ processors, equipped with a Sirius Video™ broadcast-quality video capture/display board, and a 21in standard NTSC television.[1] The front end of the eye tracker is shown in Figure 1. The subject setup includes a head/chin rest which provides limited head stability

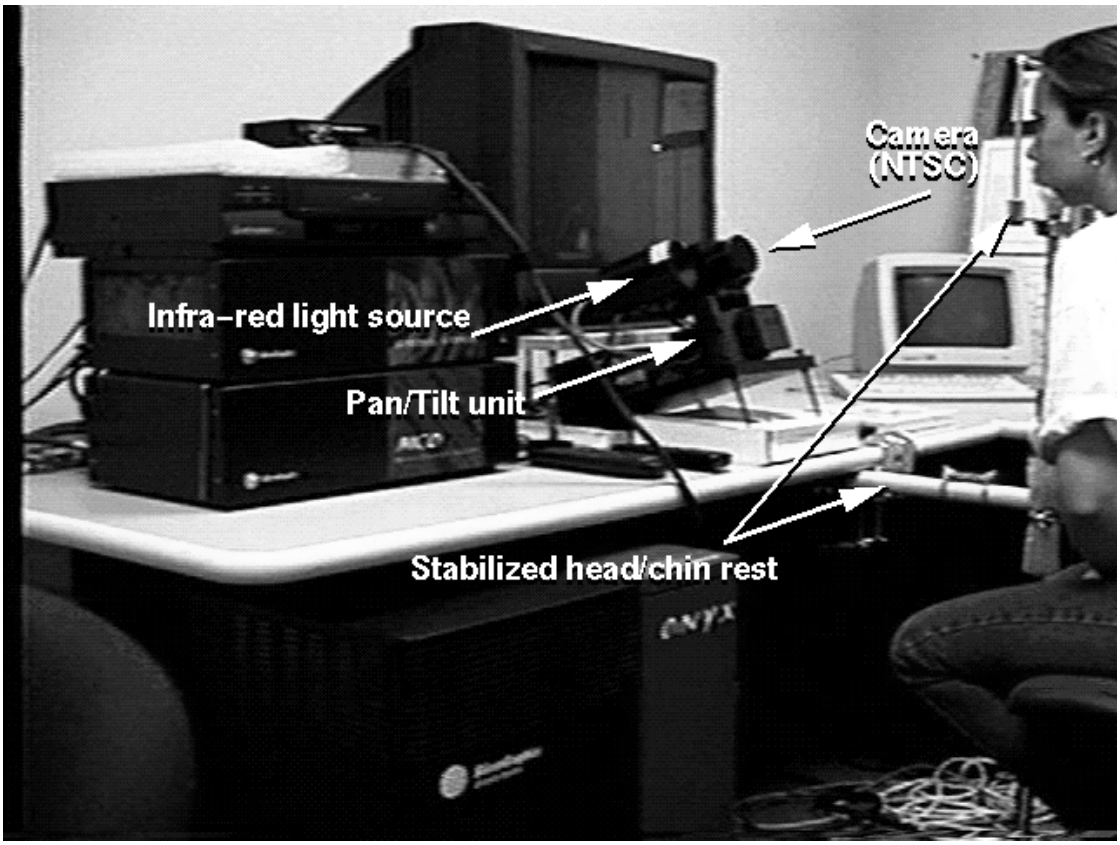


Figure 1: Virtual Environments Laboratory: eye-tracking apparatus.

while maintaining constant viewing distance for all subjects. Because the head-rest sits atop a monopod (not a tripod) base, subjects tended to tilt the head-rest while leaning on it. A simple clamp mechanism was installed to prevents this. The clamp mechanism connects the vertical head-rest monopod to the table on which the remote eye tracker rests. A long round wooden cylinder (broom handle) is used to position the head-rest relative to the television so that the eye-screen distance is maintained at about 60cm. This cylinder is fastened to the vertical head-rest pole at one end, and to a flat block of wood at the other. Butterfly (wing) nuts are used to allow positioning of the horizontal cylinder. Once measured, the assembly is tightened to stabilize the head-rest.

The video eye tracker is composed of a desk-top digital camera and infrared light source connected to a dedicated

[1] Silicon Graphics, Onyx, RealityEngine2, are registered trademarks of Silicon Graphics, Inc. Sirius Video is a trademark of Silicon Graphics, Inc. As of this writing, see URL: `http://www.sgi.com/Misc/external.list.html` for a complete list of trademarks.

personal computer (PC). Using the vendor's proprietary software and hardware, the PC calculates the subject's real-time (60Hz) fixation position from the video image of the subject's corneal reflection of the infrared light source (first Purkinje image). In the present experimental setup, the eye tracker is treated as a black box delivering real-time fixation (x, y) coordinates over a 19.2 Kbaud RS-232 serial connection. One of the inherent problems in any gaze-contingent system is the latency of the eye tracker. In the case of the ISCAN tracker, although its sampling frequency is 60Hz, the calculations required to obtain fixation position (e.g., calculations needed to disambiguate eye movements from head movements) incur an additional latency. The vendor guaranteed availability of an updated fixation data word within a period not exceeding 18ms.

The Sirius Video™ subsystem provides broadcast-quality video capture and playback. Standard NTSC video is captured from standard VHS video running on an off-the-shelf VCR. The NTSC video format (NTSC/component 525, CCIR square-pixel) provides 525 lines of resolution allowing 640×480 video frames.[6] The SGI software libraries provide capabilities for the transfer of video frames encoded in RGBA format, with each channel represented by 8 bits for a total of 32 bits per pixel (other formats are also available but are not currently used). An in-house program was developed to collect video frames in memory. Video frames are composed by interlacing NTSC video fields. Video playback is also facilitated through the Sirius board and an in-house program. Video frames are bisected into NTSC video fields before initiating memory-to-video transfer.

The effective video display rate is constrained by the availability of processors and the number of concurrently competing processes. In order to meet the NTSC display rates the original target rate was set to 30fps. Testing has shown that 30fps display rates are possible provided no other directly competing processes exist simultaneously on the machine. If, however, concurrent processes are competing for CPU resources, the display rate drops dramatically. In the case of the eye tracking system, a process concurrent with the video display routine is required to sample the serial port. Due to the competitive nature of these dual processes, both objectives of target sample and display rates could not be met simultaneously. Instead, a balance was sought where motion in the video sequence could still be perceived while eye tracker data was obtained at the fastest rates possible (see §5.2). The multithreaded system developed for this task is described in §3 below.

3 SOFTWARE

Experiments were carried out in the Virtual Environments Laboratory. The laboratory environment, with an operator and subject present, is shown in Figure 2. The gaze-contingent video (`gcv`) display system was developed to simultaneously display video and obtain raw eye movement data. The system evolved over several revisions. Although initial attempts were developed independently, the final version in many ways parallels the virtual environment system described by Jacoby et al.[2] The implementation of the multi-process, shared memory `gcv` system evolved for very similar reasons (common hardware and software platforms), and in the case of shared memory, with the help of one of the authors. The minimization of end-to-end latency and effective update rate was the common driving force behind `gcv`'s and Jacoby et al.'s designs.

Eye movement sample and video display rates of the `gcv` system are measured by taking time-stamp differences at specific points in the source code during each cycle of the time critical software loops.[2] Cycle times (periods)

[2]Preliminary versions of the system used the C callable function `gettimeofday` which, according to the system manual pages, has a resolution of 1ms. Examination of various SGI users' comments and critiques of this utility on various SGI-related discussion newsgroups (e.g., `comp.sys.sgi.bugs`, `comp.sys.sgi.graphics`, `comp.sys.sgi.hardware`, `comp.sys.sgi.misc`) revealed that this mechanism may be unreliable and that its resolution may in fact be as low as 10ms. Finding this inadequate, an alternative time-stamp `dmGetUST`, part of the Digital Media Library™, was used instead as suggested by various SGI users. This time-stamp is maintained in SGI hardware and has a reported nanosecond resolution (it is a 64-bit number representing nanoseconds since the last system reboot). It has been found more than adequate for millisecond timing purposes although it is non-portable.

are inverted (i.e., 1/period = rate) yielding update rates for that software loop. In the case of the video display, a data structure was created which maintains the index of the currently displayed frame. This frame counter advances as soon as the video update rate exceeds the desired frame rate. The averaged sampled cycle time over the duration of the displayed sequence is reported giving an effective display rate. Note that (especially) on a UNIX system, sampled cycle times need to be averaged to account for the underlying stochastic variation of the observed context-switchable process.[2] The sampling rate of the eye tracker data collection component is obtained

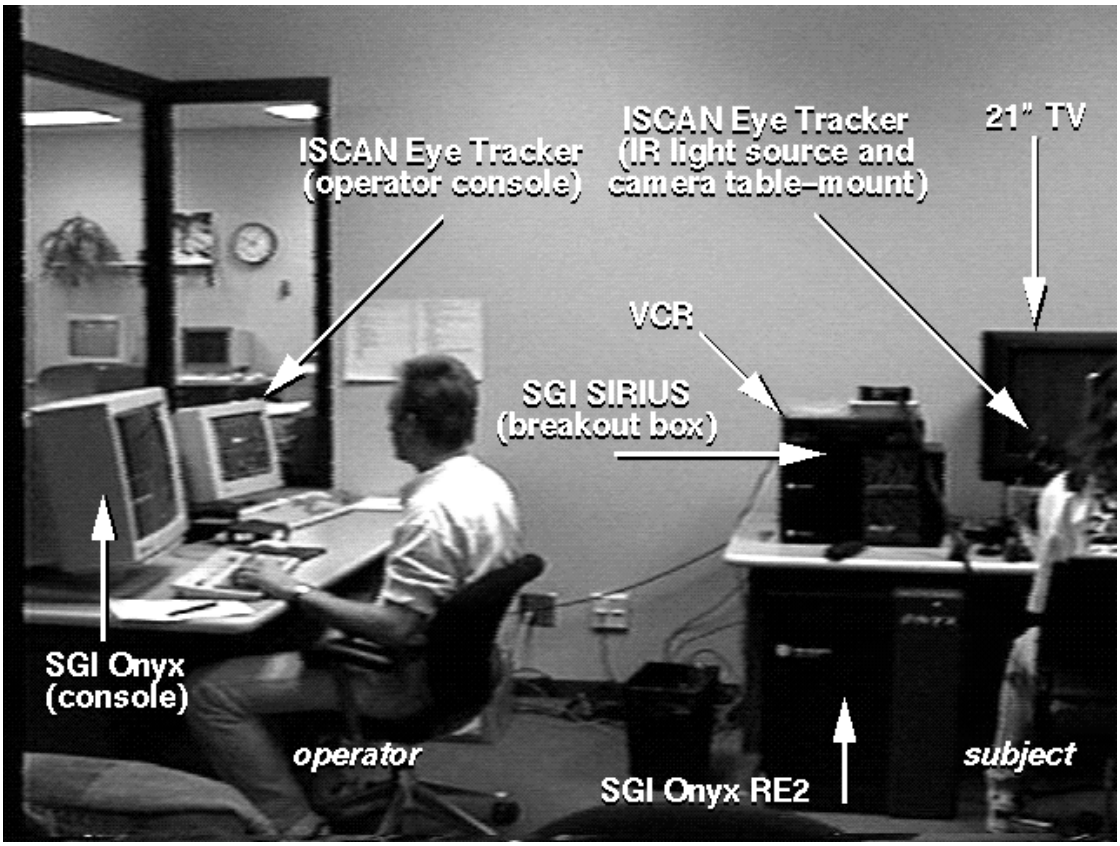


Figure 2: Virtual Environments Laboratory: laboratory setup.

similarly. In this case the next eye movement tuple (x, y) is not collected from the serial port buffer until the desired period has elapsed.

The `gcv` system organization was broken into seven main sub-processes (threads, or light-weight processes) to take advantage of the operating system's (IRIX™ version 5.3) real-time library extensions (via arguments to the system calls `sysmp` and `schedctl`). During development, it was found that the bottleneck of the system was the transfer of a video frame from memory to the Sirius board. This transfer is accomplished through a sequence of calls to the Video Library (`vlGetNextFree, vlGetActiveRegion`), the memory transfer (`memcpy`), and a final Video Library call (`vlPutValid`). Although the system manual pages offer `memcpy` as the fastest available memory transfer method, due to the voluminous nature of video data, this point in the video transfer loop was identified as the source of congestion. In early stages of development, when the `gcv` system was run as a single process, this video transfer delay prevented timely collection of eye movement data from the serial port buffer. Separating the competing software modules into concurrent process threads allowed more frequent context switching, thereby allocating system resources to the threads more fairly. A further gain in update rate performance was achieved by locking the video transfer process onto its own processor. Since the eye tracker data collection process requires a much smaller amount of data transfer (on the order of bytes), it was locked together

with the remaining bookkeeping threads on the other available processor. Finally, all processes were given the same non-degrading priority to ensure fair treatment. In all, the following six sub-processes were spawned by the parent process via the `sproc` system call:

`MAIN` The main (parent) process: spawns 6 child processes and waits for the quit signal.

`VID` The video bookkeeping thread: in charge of maintaining video status, e.g., current frame number being displayed, video play, video stop, etc. Signals process `DRW`.

`TRK` The tracking thread: in charge of maintaining tracking status (number of samples, sample rate, etc.) and the recording of eye movement data, known as the current Point Of Regard (POR). Signals process `DSC`.

`DRW` The drawing thread: solely in charge of signaling processes `DVD` and `DSC`.

`DVD` The draw-to-video thread: initiates memory to video transfer.

`DSC` The draw-to-screen thread: initiates screen update. This thread displays the current video frame on the Onyx monitor with an overlay of the current POR. Although this feature is visually informative for the experimenter, due to the video-to-framebuffer transfer, it is also a point of congestion. It is mostly used as a debugging aid and is turned off during experimentation.

`GUI` The graphical user interface thread: maintains vigil over user inputs.

Thread synchronization is achieved through the use of semaphores, provided by UNIX intrinsic system function calls (`semget, semctl, semop`). The `gcv` software system organization is shown in Figure 3, where semaphores are identified by a symbol representing a small flag. The raised flag symbolizes a semaphore signal operation,

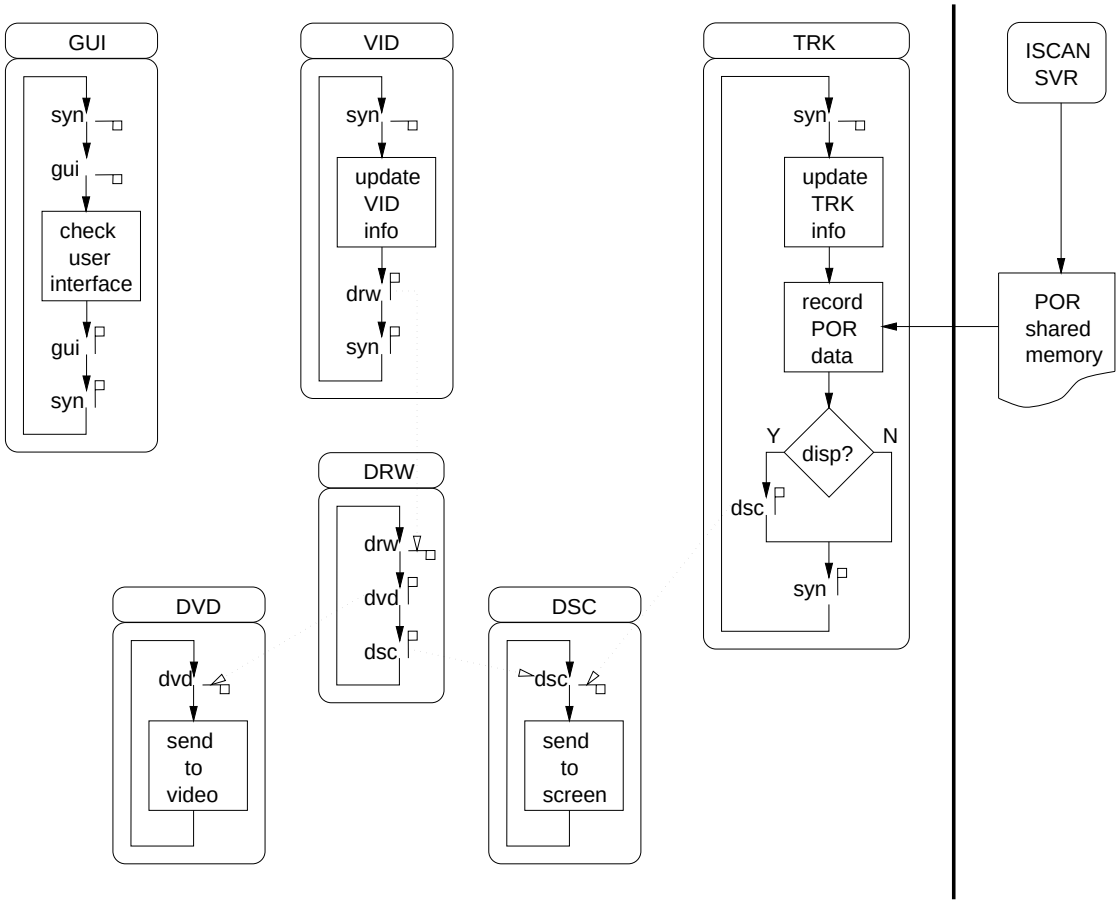


Figure 3: Eye tracking software system organization.

while the lowered flag depicts a semaphore wait. Semaphores are named after the process that they control,

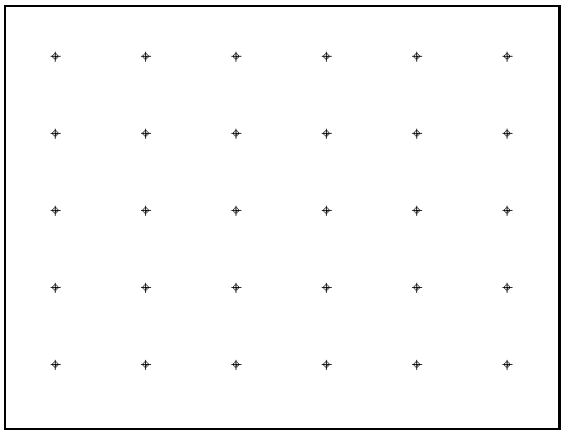
except for processes `GUI`, `VID` and `TRK`, which are controlled by the common semaphore `syn`.

From the experiences provided by Jacoby et al.,[2] it was decided to dissociate the RS-232 serial port driver process from the `gcv` system. The result is the `svr` program which is solely responsible for accessing the serial port, reading the raw eye movement data, and updating a segment of shared memory from which the `gcv` system obtains the raw POR data. The `gcv` system obtains a pointer to the shared memory segment by first issuing the system call `shmget` with an agreed key to obtain the proper identification, and then by calling `shmat` to attach the pointer. Access to the memory is controlled by a semaphore previously created by `svr`. In effect, `svr` acts as a server/writer process always writing to the shared buffer, and `gcv` acts as the client/reader process always reading from the shared buffer. The `svr` program is responsible for destroying the semaphore and shared memory segments. This protocol requires that `svr` must be running prior to starting the `gcv` system.

4 CALIBRATION PROCEDURES

Each experimental trial includes three calibration steps. Calibration is divided into two procedures: *external* and *internal*. External calibration pertains to the proprietary eye tracker instrument calibration procedure specified by the manufacturer. Internal calibration pertains to the procedure developed within the `gcv` system in order to measure eye tracker accuracy. The eye tracker is externally calibrated using the vendor's proprietary 9-point calibration procedure. `Gcv` internal calibration was developed over 30 points. External calibration points were positioned to match internal outliers. The layout of the internal calibration points denoted by the symbol + is shown in Figure 4(a). The point arrays are framed for clarity, horizontal and vertical lines do not appear during calibration. Figure 4(b) shows the position of external calibration points (depicted by circles) overlayed on top of internal calibration points. The location of calibration points is described in two coordinate systems: the

(a) Internal calibration points.



(b) External (eye tracker) calibration points overlayed over internal calibration points.

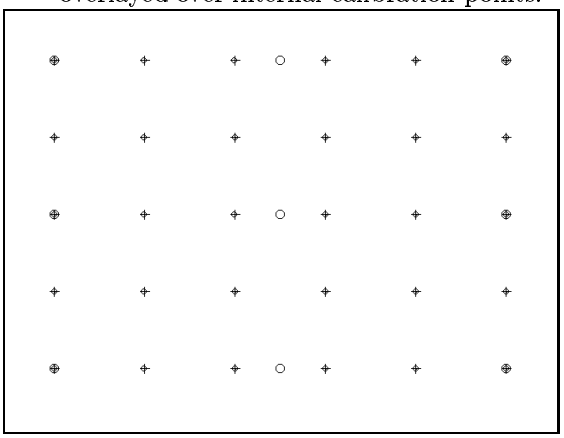


Figure 4: Calibration stimulus.

internal (`gcv`) image coordinate space, and the external (eye tracker) coordinate space. Computation of a mapping transformation from (external) eye tracker coordinates to (internal) image coordinates is described in §4.2. The following description of the layout of both internal and external calibration points is in image coordinate space.

The description of the calibration point layout is based on a viewing distance of 60cm (23.622in). Viewing distance is maintained by a stabilized head/chin rest described above. Using television resolution specifications (NTSC standard), the stimulus dimensions are obtained from the following Pythagorean equations,

$$\frac{x}{y} = \frac{640}{480} = \frac{4}{3}; \quad x^2 + y^2 = 21^2$$

where solving for the unknowns gives $y = 12.6$in., and $x = 16.8$in. Denoting the length of the base of the visual

angle by r, the horizontal visual angle θ_x subtended by the subject is given by

$$\theta_x = 2\tan^{-1}\left(\frac{r}{2D}\right) = 2\tan^{-1}\left(\frac{16.800}{47.244}\right) \doteq 39.2°,$$

where $D = 23.622$ is the viewing distance in inches. The vertical visual angle θ_y subtended by the subject is obtained similarly,

$$\theta_y = 2\tan^{-1}\left(\frac{r}{2D}\right) = 2\tan^{-1}\left(\frac{12.600}{47.244}\right) \doteq 29.9°.$$

The effective resolution in dots per inch (dpi) of the television is found by dividing the number of pixels by the monitor dimensions,

$$\frac{640}{16.8} = \frac{480}{12.6} \doteq 38\text{dpi}.$$

The internal calibration point distribution is based on a 108 horizontal and 92 vertical pixel displacement starting at the top-left pixel location (50,50). The bottom-right calibration point is located at (590,418). At the above resolution, the horizontal inter-point distance is

$$r_x = \frac{108 \text{ pixels}}{38 \text{ dpi}} = 2.842\text{in},$$

resulting in the horizontal inter-point subtended visual angle of about 7° as derived below:

$$\theta_x = 2\tan^{-1}\left(\frac{r}{2D}\right) = 2\tan^{-1}\left(\frac{2.842}{47.244}\right) \doteq 6.88° \approx 7°.$$

Similarly, the vertical calibration point displacement of 92 pixels results in a subtended visual angle of about 6°.

Internal calibration is performed twice, immediately after external calibration (before stimulus display), and immediately after the stimulus display. Hence, internal calibration is used to check the accuracy of the eye tracker before and after the stimulus display, to check for instrument slippage. The 30 internal calibration points are shown in random order in a semi-interactive manner similar to the external calibration procedure. As each point is drawn on the screen (the subject is presented with an × to minimize aliasing and flicker effects of the analog display), the `gcv` system waits for a key input before sampling eye movement data for a period of 800ms. The input key delay allows the operator to observe eye stability on the eye tracker's eye monitor. The eye is judged to be stable once the eye tracker has repositioned the eye image in the center of the camera frame. Recorded point of regard data is mapped to image coordinates in real-time. Calibration data is stored in a flat text file for later evaluation.

4.1 Stimulus Display Distortion

Video data presented on the television display is subject to a geometric transformation due to several possible optical distortions present in the display.[5] Although no severe aberrations such as those described by Robinett and Roland were observed in the current television display, internal calibration points appeared slightly displaced from expected locations due to the curvature of the picture tube. This distortion is known as the pin-cushion effect.[1] Pixel data is effectively spread out concentrically from the center of the screen, most noticeably in the corners of the display. The global effect is hardly noticeable, especially in viewing imagery, but local pixel perturbations are significant. To illustrate, the upper left calibration point at location (50,50) in image coordinates is displayed at a location near (21,21) in television coordinates.[3]

One possible compensation method for the optical nonlinearity of the display is a predistortion of the stimulus image. For example, to correctly display a straight line in the internal image, a curved line needs to be drawn

[3]The TV measurement is an approximate count of the television's RGB pixel triads.

on the external device, balancing out the optical distortion. This method is computationally expensive since it needs to be performed over the entire image. In the present eye tracking application, it is more important to obtain correct POR measurement instead of ensuring undistorted display of the stimulus. That is, instead of a predistortion transformation of the television input, a suitable method is sought for real-time, point-by-point correction of the eye tracker output.

The pin-cushion effect can be compensated for automatically, if an appropriate transformation can be derived from the eye tracker's measurements of displayed pixel locations. That is, if eye tracker coordinates can be passed through the same display distortion as the image coordinates, e.g., pixel information from both sources appearing on the same display device, the distortion effects will effectively cancel. This is the motivation behind the eye tracker data transformation described below.

4.2 Eye Tracker Coordinate Transformation

Two different coordinate spaces represent stimulus imagery and gaze position, respectively. Dimensions of stimulus imagery are based on the dimensions of the video display (640×480 pixels) while gaze position is dependent on the resolution of the eye tracker (512×256 pixels). Since gaze position information is sought in image coordinates, a transformation is required mapping eye tracker coordinates to image coordinates. The mapping is graphically depicted in Figure 5, where image and eye tracker dimensions are shown approximately to scale.

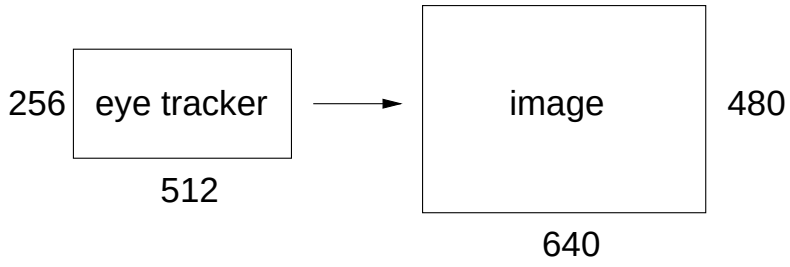


Figure 5: Eye tracker-image coordinate space mapping transformation.

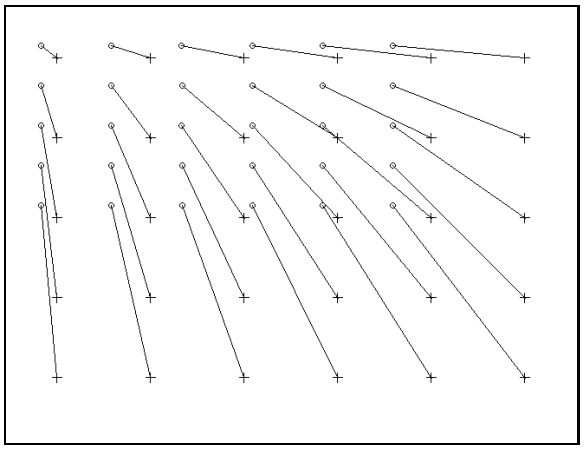
Eye tracker coordinate transformation is facilitated by the eye tracker's fine-grained cursor position readout to estimate the coordinates of the displayed calibration points. Since the video signal is sent through the eye tracker, the calibration points are displayed on the eye tracker stimulus monitor. The eye tracker cursor is then positioned over the calibration points. A status readout on the monitor shows the cursor's x- and y-coordinates in eye tracker coordinate space. Calibration points as measured by the eye tracker are shown in Figure 6(a), where points in eye tracker space are represented by circles joined with a line to the corresponding point in image space represented by a +. The point arrays are framed for clarity, horizontal and vertical lines do not appear during calibration. The apparently large discrepancy is a result of the eye tracker's limited vertical resolution (approximately half the image height).

Corrected POR data in image space, (x', y'), is obtained from raw POR data, (x, y), through the transformation,

$$x' = 7.58 + 1.33x, \quad y' = -22.00 + 2.00y.$$

The translation and scale factors correspond to the dimensions and relative spatial location of eye tracker and image coordinates. In practice, a restricted subset of the eye tracker coordinate space was used to estimate corresponding locations of calibration points in image space. The scale value of 1.33 corresponds to the scale factor between the eye tracker and image x-coordinates, $[32, 438]$ and $[50, 590]$, respectively. The dimension ratio $(590 - 50)/(438 - 32)$ gives 1.33. The y-coordinate scale value of 2.00 is obtained given eye tracker and image y-coordinate ranges, $[36, 220]$ and $[50, 418]$, respectively. Translation factors may be similarly derived from the relative location of the upper-left corners of the coordinate spaces, shown overlayed, approximately to scale in Figure 6(b).

(a) Eye tracker-image space measurements.



(b) Eye tracker-image space overlay.

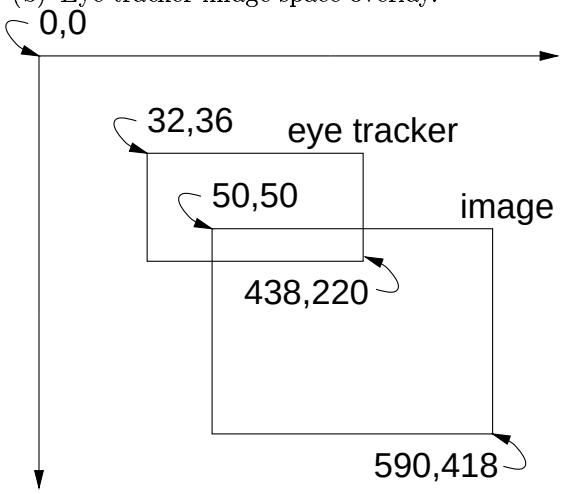


Figure 6: Eye tracker-image coordinate transformation.

5 RESULTS

Eye tracking experiments were carried out in the Virtual Environments Laboratory, Department of Computer Science, Texas A&M University. In this section, performance of the present system is reported in terms of (1) eye tracker sampling latency and video transfer rates, and (2) measured eye tracker accuracy and slippage. Results are reported from a preliminary set of experiments involving 7 subjects. Although equally important, gaze position error is not considered here since it does not bear significance on the evaluation of the eye movement recording strategy (but see reports of subsequent experiments where gaze position is measured during peripheral image manipulation dependent on viewers' collocation of gaze with intended regions of interest[?]). Eye tracker accuracy was measured by internal calibration procedures described in §4. Measurements were taken before and after stimulus viewing trials. These procedures provide the basis for two statistical measures: (1) the overall accuracy of the eye tracker, and (2) the amount of instrument slippage during stimulus viewing. The latter measurement gives an indication of the instrument accuracy during the viewing task, i.e., by recording loss of accuracy between the before- and after-viewing calibration procedures.

5.1 Experimental Trials

Each experimental trial consists of three video presentations to an individual. Trials are limited to three video presentations since loading of the video into memory requires 7 minutes and each trial is designed to process an individual in roughly 30 minutes. Each trial is composed of the following steps:

1. Brief introduction. When a subject enters, s/he is asked to sit in front of the eye tracker head rest. The equipment is briefly described with emphasis on the eye tracker infra-red (IR) light source and camera. It is pointed out that the IR assembly contains a standard overhead projector bulb as the light source. This is done to alleviate any preconceived fears regarding the apparatus (some subjects were under the impression that eyelid movement would be restricted). Subjects are assured that the experiment is physically unobtrusive.
2. A training video may be shown prior to experimentation, depending on the experimental design requirements.
3. Video presentation. Each video presentation consists of the following substeps:
   (a) External calibration. Once the subject has settled into the head rest, the eye tracker is calibrated, as discussed in §4.
   (b) Internal calibration. Immediately following external calibration, the internal calibration procedure is

performed to record the initial accuracy of the eye tracker.

(c) Stimulus display. The video is presented twice in succession each time eye movements are recorded and stored.

(d) Internal calibration. Immediately following stimulus presentation, the internal calibration procedure is performed once again to record the final accuracy of the eye tracker.

4. Break. The subject is told to relax while the next video sequence is loaded. During this break, the subject may fill out a questionnaire, informed consent forms, etc. Step 3 is then repeated.

5.2 Eye Tracker Sampling Latency and Video Transfer Rates

To afford a fast eye movement sampling rate, the target display was decreased from 30fps to 15fps to match minimum motion perception requirements of the human visual system. Frame rates as low as 5fps have been suggested as a critical minimum rate for acceptable subjective quality (in the context of audio enhanced video conferencing).[4] The critical interstimulus interval (i.s.i.) range required for the hybrid perception of stroboscopic and continuous motion is approximately 32-64ms (30-15fps).[3] This range corresponds to the two qualities of human vision responsible for seamless perception of television imagery: the *critical fusion frequency* of about 30Hz which allows flicker-free perception of strobe-like 30fps NTSC video (60 interlaced fields per second), and the perception of *apparent motion* of spatially displaced objects with i.s.i of roughly 64ms (15fps). Testing has shown that a maximum sustained display rate of 16fps is achievable in conjunction with eye tracker data collection at (approximately) 60Hz.

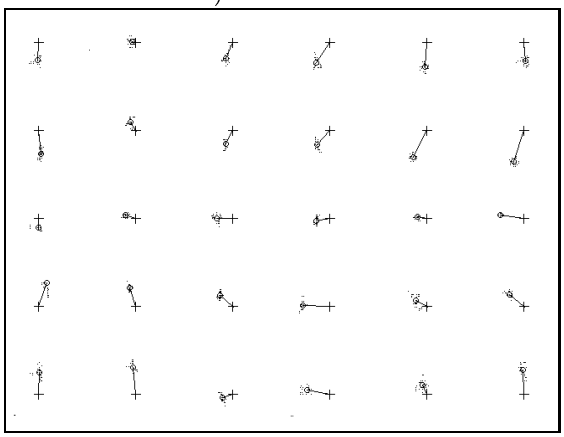
The length of video sequences for recording and playback (to and from memory) is limited by the relatively small amount of RAM. The Onyx computer is currently equipped with 320 megabytes of base memory. With each pixel represented by 4 bytes, one video frame requires approximately 1.23 megabytes of storage. Theoretically, approximately 260 frames of uncompressed video may be stored in RAM. Compared to RAM, disk access is extremely slow and must be avoided in order to ensure fast video display rates. Since much of the available memory is consumed by the operating system and the user programs required to control video display, experience has shown that a maximum of 128 NTSC (640×480) video frames may be stored in RAM before disk swaps are required, providing 8 seconds of stimulus duration. Since the primary goal of the experiments was the testing of a gaze-contingent variable resolution display strategy,[?] monochrome (greyscale) video was used.

5.3 Eye Tracker Accuracy

Eye tracker readings were obtained over 30 internal calibration points as described in §4. Each calibration point measurement consists of eye tracker samples about the calibration point over an 800ms period (approximately 44 individual data points). Raw sample points falling in the exterior 10-pixel wide borders are ignored. This is due to the eye tracker's property of generating (0,0) values during blinks (confirmed by the vendor). For this reason, any time a raw sample point is close enough to the location (0,0) (within 10 pixels), it is removed from further consideration. An average of valid data points (centroid) is obtained and the error between the centroid and calibration point is calculated. Each two-dimensional euclidian distance measurement is converted to the full visual angle dependent on the viewing distance and calculated resolution of the television screen. Thus each calibration run contains 30 average measured deviations at each calibration point in terms of visual angle. A graphical example of this measurement is shown in Figure 7. The internal calibration locations are represented by +, sample measurements are represented by individual pixel dots, and centroid gaze positions are represented by circles, joined with the corresponding calibration point by a line. The length of the line is the average error deviation in pixels.

To quantify the overall eye tracker accuracy succinctly, the average calibration error is obtained from each set of calibration points in order to calculate an overall average statistic of the eye tracker. The resulting average instrument error is an average statistic over all calibration runs. The calculated mean value is 1.87°. This is not a particularly informative statistic since the data does not appear to fit a normal distribution. Since the

(a) Calibration before stimulus viewing (avg. error: 1.40°).



(b) Calibration after stimulus viewing (avg. error: 1.77°).

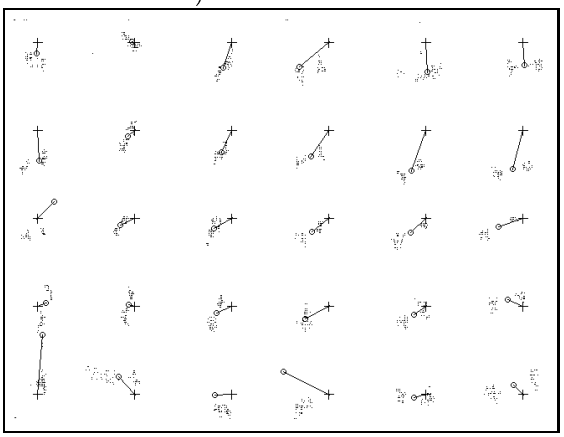


Figure 7: Typical per-trial calibration data (subject # 21).

average error data appears skewed, a more meaningful statistic is the median value, which ignores the influence of outliers. Its value is 1.41°. Using similar reasoning for reporting a dispersion statistic, the interquartile range (iqr) is utilized instead of the standard deviation for its robust response to outliers. The iqr value is 0.97°. These findings indicate an overall acceptable performance, not far off from the vendor's claimed accuracy (roughly 1°visual angle).

5.4 Eye Tracker Slippage

Quantification of the before- and after-viewing eye tracker error provides a measure of instrument accuracy during the viewing task. A graphical example of this measure is shown in Figure 8 which is a composite plot of Figures 7 (a) and (b). Notice that measured eye positions in relation to calibration points coincide well overall. To quantify

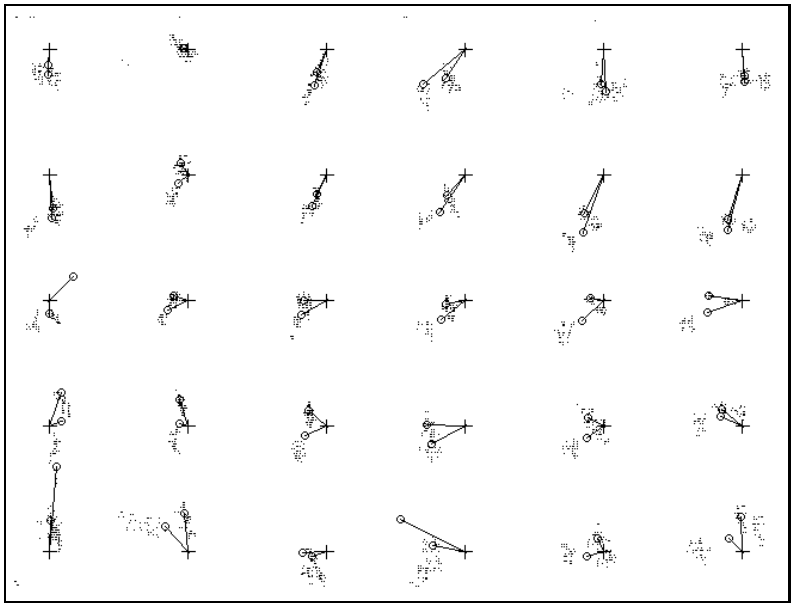


Figure 8: Composite calibration data showing eye tracker slippage (subject # 21).

this correspondence, a one-way ANOVA was performed on the means of the before- and after-viewing average error measures. On average, no significant slippage is detected by this statistic. Note that ANOVA in this case is not very informative since it does not consider eye tracker slippage on a per-trial basis. That is, the ANOVA only reports significant correspondence of the mean measurements.

To examine eye tracker slippage on a per-trial basis, differences of average errors were calculated between the before- and after-viewing calibration runs on a per-run basis. Judging by the overall difference error histogram, difference measurements appear to fit a skewed distribution, again calling for the use of statistical measures robust to outliers. The median error is -0.25°, and interquartile range is 0.40°. These values quantify the close correspondence of example pre- and post-viewing calibration measurements shown graphically in Figure 8. Overall, the tracker accuracy varies roughly a quarter of a degree visual angle over the 8-second viewing task. Over some calibration points, accuracy improves while over others it degrades. Since the peak frequency is close to 0, generally the accuracy before and after viewing the stimulus remains stable overall.

6 CONCLUSION

Eye tracking experiments were carried out in the Virtual Environments Laboratory, Department of Computer Science, Texas A&M University. Experimental results are provided describing the `gcv` system's performance in terms of (1) eye tracker sampling latency and video transfer rates, and (2) measured eye tracker accuracy and slippage. Optimal system performance was empirically achieved with an (average) 18ms sampling latency at sustained video transfer rates of 16fps. Eye tracker accuracy is measured over 30 calibration points before and after stimulus viewing trials. The median error of all measurements was 1.41° (visual angle). Eye tracker slippage was calculated as the difference in error measurements between pre- and post-viewing calibration runs. The overall median slippage error was -0.25° (visual angle). Negative values suggest slight instrument slippage.

The observed eye tracker accuracy and slippage indicate an overall acceptable performance, not far off from the vendor's claimed accuracy (roughly 1° visual angle). The programming strategies developed for incorporating the viewer's POR in virtual enviroment systems are independent of proprietary eye tracking equipment and are applicable to general gaze-contingent virtual environment and gaze-contingent virutal reality (GCVR) designs.

7 REFERENCES

[1] James D. Foley and Andries van Dam. *Fundamentals of Interactive Computer Graphics.* Addison-Wesley, Reading, MA, 1982.

[2] Richard H. Jacoby, Bernard D. Adelstein, and Stephen R. Ellis. Improved temporal response in virtual environments through system hardware and software reoganization. In Mark T. Bolas and Scott S. Fisher, editors, *Stereoscopic Displays and Virtual Reality Systems III*, San Jose, CA, January 30-February 2 1996. SPIE.

[3] M. J. Morgan. Analogue models of motion perception. *Phil. Trans. R. Soc. Lond. B*, 290:117–135, 1980.

[4] Thrasyvoulos N. Pappas and Raynard O. Hinds. On video and audio data integration for conferencing. In *Human Vision, Visual Processing, and Digital Display VI*, pages 120–127, San Jose, CA, February 1995. SPIE.

[5] Warren Robinett and Jannick P. Rolland. 5: A Computational Model for the Stereoscopic Optics of a Head-Mounted Display. In R. A. Earnshaw, M. A. Gigante, and H. Jones, editors, *Virtual Reality*, pages 51–75. Academic Press, London, 1993.

[6] Silicon Graphics Inc. (SGI), Mountain View, CA. *IRIS Media Libraries™ Programming Guide*, March 1995. Document Number 007-1799-030.