Binocular Eye Tracking in VR for Visual Inspection Training

Andrew T. Duchowski* Computer Science
Eric Medlin* Computer Science
Anand Gramopadhye† Industrial Engineering
Brian Melloy† Industrial Engineering
Santosh Nair † Industrial Engineering
Clemson University

ABSTRACT

This paper presents novel software techniques for binocular eye tracking within Virtual Reality and discusses their application to aircraft inspection training. The aesthetic appearance of the environment is driven by standard graphical techniques augmented by realistic texture maps of the physical environment. The user's gaze direction, as well as head position and orientation, are tracked to allow recording of the user's fixations within the environment. Methods are given for (1) integration of the eye tracker into a Virtual Reality framework, (2) stereo calculation of the user's 3D gaze vector, (3) a new 3D calibration technique developed to estimate the user's inter-pupillary distance *post-facto*, and (4) a new technique for eye movement analysis in 3-space. The 3D eye movement analysis technique is an improvement over traditional 2D approaches since it takes into account the 6 degrees of freedom of head movements and is resolution independent. Results indicate that although the current signal analysis approach is somewhat noisy and tends to underestimate the identified number of fixations, recorded eye movements provide valuable human factors process measures complementing performance statistics used to gauge training effectiveness.

1. INTRODUCTION

Interest in eye tracking interface techniques has endured since early implementations of eye-slaved flight simulators and has since permeated several disciplines including human-computer interfaces, teleoperator environments, and visual communication modalities [12, 16, 9]. Advancements in eye tracking technology, specifically the availability of cheaper, faster, more accurate and easier to use trackers, have inspired increased interdisciplinary eye movement and eye tracking research efforts. A good example of this type of diverse interest can be found in the proceedings of the recent Eye Tracking Research & Applications Symposium [4].

Recent applications of an eye tracker in Virtual Reality (VR) have shown promise in the use of the device as a component of multi-modal interface systems. For example, Tanriverdi and Jacob have used an eye tracker as a selection device in VR [17], and Danforth et al. use an eye tracker as an indicator of gaze in a gaze-contingent multi-resolution terrain navigation environment [2]. These are examples of eye trackers used for *interactive* needs. In the present application, the eye tracker serves a *diagnostic* purpose.

In contrast to interactive applications, diagnostic eye tracking systems typically involve the recording of eye movements over time for analysis of the user's overt visual attention over a given stimulus. No changes in the display occur due to the location of the user's gaze, rather, the user's eye movements are simply (and unobtrusively) recorded in real-time for post-immersion analysis. Numerous examples of general diagnostic applications can be found (see for example [6]), however, diagnostic eye tracking methods in VR have not been widely adopted.

2. BACKGROUND

Aircraft inspection and maintenance are an essential part of a safe, reliable air transportation system. Training has been identified as the primary intervention strategy in improving inspection performance [8]. If training is to be successful, inspectors need to be provided with training tools to help enhance their inspection skills. In response to this need, this paper describes a diagnostic eye tracking VR system developed for the purpose of recording process measures (head and eye movements) as well as performance measures (search time and success rate) during immersion in a VR aircraft inspection simulator. The VR simulator features a binocular eye tracker, built into the system's Head Mounted Display (HMD), which allows the recording of the user's dynamic Point Of Regard (POR) within the virtual environment. Although binocular eye trackers integrated with HMDs have previously been proposed [14], no reports of their actual construction or operation have been found.

The purpose of this paper is to provide technical details of the eye tracking software techniques developed for both cognitive feedback and collection of users' process measures aiding post-immersive performance evaluation. User gaze directions, as well as head position and orientation, are tracked to enable post-immersive examination of the user's overt spatio-temporal focus of attention while immersed in the environment. Recorded eye movements address imprecision and ambiguity of the user's viewpoint in VR by explicitly providing the 3D location of the user's gaze. The collection of gaze points taken over the course of immersion (i.e., the user's three-dimensional scanpath), serves as a diagnostic tool for post-immersive reasoning about the user's actions in the environment.

*{andrewd | emedlin}@vr.clemson.edu
†{agramop | bmelloy | santosn}@ces.clemson.edu

VRST'01, November 15-17, 2001, Banff, Alberta, Canada.

3. THE VIRTUAL ENVIRONMENT

The VR inspection system is a collaborative extension of recent efforts focusing on the development of a computer based inspection training program. The resulting program, developed using a task analytic methodology, features a PC-based inspection simulation of an aircraft cargo bay, where an image of a portion of the airframe is presented to the user for inspection (visual detection of defects). Despite its advantages of being a computer-based inspection training/job-aid tool, the static, two-dimensional layout of the airframe lacked realism. To enhance the fidelity of the inspection system, an immersive, three-dimensional VR system has been developed. The goal of the construction of the virtual environment is to match the appearance of the physical inspection environment, an aircraft cargo bay, shown in Figure 1. The physical environment

Figure 1: Aircraft cargo bay physical environment.

is a complex three-dimensional cube-like volume, with airframe components (e.g., fuselage ribs) exposed for inspection. A typical visual inspection task of the cargo bay involves carefully walking over the structural elements while searching for surface defects such as corrosion and cracks.

The model of the virtual inspection environment was patterned after a simple three-dimensional enclosure (e.g., a cube), specified by the dimensions of the real inspection environment (i.e., an aircraft's cargo bay). The model is built entirely out of planar polygons. There are two pragmatic reasons for this design choice. First, since the representation of the true complexity of the airframe structure is avoided, fast display rates are possible. Second, planar polygons (quadrilaterals) are particularly suitable for texture mapping.

While our graphical environment is relatively simple, it appears to be sufficiently realistic for the purposes of inspection training. An experiment conducted to evaluate the subjective quality of the simulator attempted to measure the degree of presence felt by participants immersed in the environment [19]. Analysis of responses to a modified version of Witmer and Singer's Presence Questionnaire [21] revealed that the system scored high on presence-related questions. Visual aspects of the environment, sense of objects, anticipation of system response, surveying, and experience in the environment all contributed to a reported high level of involvement in VR. Although student subjects were not qualified inspectors, on average, they indicated their experience in the virtual environment to be consistent with a walkthrough of a real aircraft prepared for inspection. We expect trained inspectors will find the simulator similarly consistent with the real environment, at least in the context of simulating the visual search task. That is, we realize that our simulator is not necessarily *photo-realistic* (e.g., due to limited resolution of the HMD, coarse and flat appearance of texture maps), however, since the purpose of the simulator is to train search behavior, we believe the simulator is sufficiently *functionally realistic* for this purpose.

4. HARDWARE COMPONENTS

Our primary rendering engine is a dual-rack, dual-pipe, Silicon Graphics Onyx2® InfiniteReality2™ system with 8 raster managers and 8 MIPS® R12000™ processors, each with 8MB secondary cache.[1] It is equipped with 8Gb of main memory and 0.5Gb of texture memory.

Multi-modal hardware components include a binocular eye tracker mounted within a Virtual Research V8 Head Mounted Display. The V8 HMD offers 640×480 pixel resolution per eye with individual left and right eye feeds. HMD position and orientation tracking is provided by an Ascension 6 Degree-Of-Freedom (6DOF) Flock Of Birds (FOB). The HMD is shown in Figure 6(a), with the FOB sensor just visible on top of the helmet. (This figure is reproduced in color on page 193.) A 6DOF tracked, hand-held mouse provides a means to represent a virtual tool for the user in the environment.

The eye tracker is a video-based, corneal reflection unit, built jointly by Virtual Research and ISCAN. Each of the binocular video eye trackers is composed of a miniature camera and infrared light sources, with the dual optics assemblies connected to a dedicated personal computer (PC). The ISCAN RK-726PCI High Resolution Pupil/Corneal Reflection Processor uses corneal reflections (first Purkinje images) of infra-red LEDs mounted within the helmet to measure eye movements. Figure 6(b) shows the dual cameras and infra-red LEDs of the binocular assembly. (This figure is reproduced in color on page 193.) Mounted below the HMD lenses, the eye imaging cameras peer upwards through a hole cut into the lens stem, capturing images of the eyes reflected by a dichroic mirror placed behind the HMD lenses. The processor operates at a sample rate of 60Hz and the subject's eye position is determined with an accuracy of approximately 0.3 degrees over a ±20 degree horizontal and vertical range using the pupil/corneal reflection difference. The maximum spatial resolution of the calculated POR provided by the tracker is 512×512 pixels per eye.

The binocular eye tracking assembly allows the measurement of vergence eye movements, which in turn provides the capability of calculating the three-dimensional virtual coordinates of the viewer's gaze. Using the vendor's proprietary software and hardware, the PC calculates the subject's real-time POR from the video eye images. In the current VR configuration, the eye tracker is treated as a black box delivering real-time eye movement coordinates (x_l, y_l, t) and (x_r, y_r, t) over a 19.2 Kbaud RS-232 serial connection, and can be considered as an ordinary positional tracking device.

5. VR EYE TRACKER INTEGRATION

Raw output from the eye tracker is shown in Figure 2, where the left and right eye POR is represented by a small circle and small crosshair, respectively, superimposed by the eye tracker's scene imaging hardware. The VR scene image signal is split (via VGA active passthrough) prior to HMD input, and diverted to the eye tracker. Thus the eye tracker and HMD simultaneously display the same image seen by the user in the HMD. In addition, each scene image generated by the eye tracker contains the superimposed POR indicator and a status bar at the bottom indicating current pupil

[1] Silicon Graphics, Onyx2, InfiniteReality, are registered trademarks of Silicon Graphics, Inc.

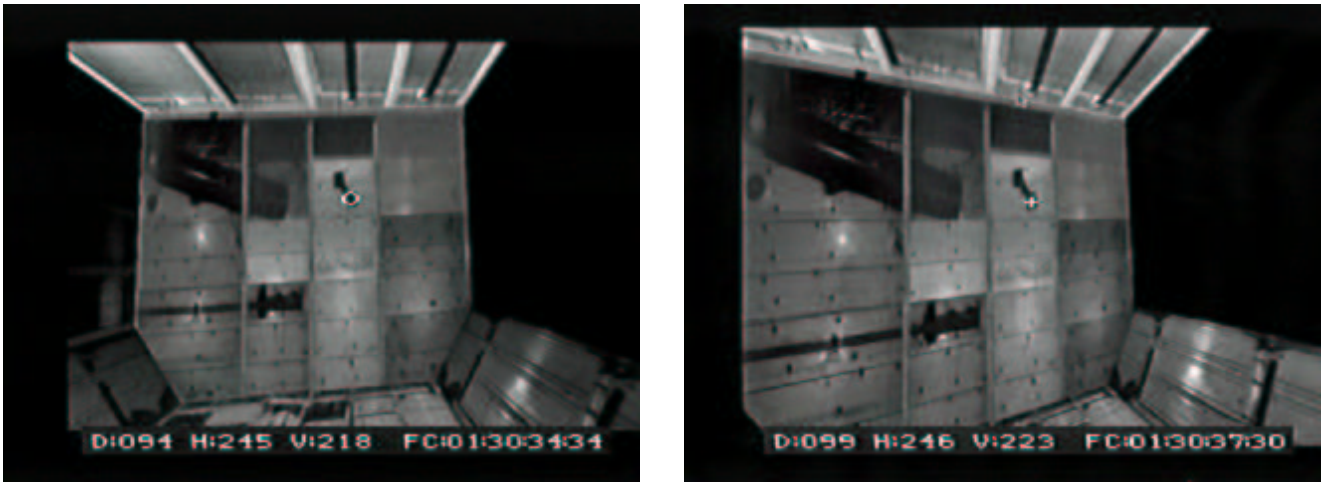


Figure 2: Raw eye tracker output: (a, left) left eye POR, (b, right) right eye POR.

diameter, horizontal and vertical POR coordinates, and the video frame counter (`HH:MM:SS:FF`). Note that the images shown in the figure were captured 3 seconds apart. Several processing steps are required to accurately calculate the user's gaze within the environment. Once the gaze direction has been obtained, the resultant gaze vector is used to identify fixated regions in the environment by first calculating the gaze/environment intersection points, then applying signal analysis techniques to identify fixations.

5.1 Eye Tracker Coordinate Mapping

A critical concern in designing a gaze monitoring VR system is the mapping of eye tracker coordinates to the application program's reference frame. The eye tracker calculates the viewer's POR relative to the eye tracker's screen reference frame (e.g., a 512×512 pixel plane, perpendicular to the optical axis). The eye tracker returns a sample POR coordinate pair for each eye. These coordinate pairs must be mapped to the extents of the application program's viewing window.

Although raw eye tracker coordinates are in the range $[0,511]$, in practice the usable, or effective, coordinates are dependent on: (a) the size of the application window, and (b) the position of the application window, both relative to the eye tracker's reference frame. Proper mapping between eye tracker and application coordinates is achieved through the measurement of the application window's extents in the eye tracker's reference frame. This is accomplished by using the eye tracker's fine cursor movement and cursor location readout.

To obtain the extents of the application window in the eye tracker's reference frame, the application window's corners are measured with the eye tracker's cursor. Figure 3(b), overleaf, illustrates an example of a 600×450 application window as it might appear on the eye tracker scene monitor. Scene monitors (one for each eye) are shown in Figure 3(a)—they are the top two small monitors, visible just under the user's left hand—the images displayed in these scene monitors were captured and shown in Figure 2. Given the extents of both application and eye tracker screen coordinates, a simple linear interpolation mapping is used to map raw POR data to the graphics screen coordinates [6]. While seemingly trivial, this mapping is key to proper calculation of the gaze vector in world coordinates from raw POR data and is also essential for alignment of target points displayed by the application program during calibration of the eye tracker. Correct registration between eye tracker coordinates and image coordinates is achieved if the linearly mapped computer-generated calibration target points align with the calibration points generated by the eye tracker. Because both coordinates are ultimately subject to the same optical distortions of the HMD (e.g., pin-cushion effect), the linear mapping is sufficient for coordinate registration [5].

5.2 Gaze Vector Calculation

The calculation of gaze in three-space depends only on the relative positions of the two eyes on the horizontal axis. The parameters of interest are the three-dimensional virtual coordinates, (x_g, y_g, z_g), which can be determined from traditional stereo geometry calculations [10]. Figure 4 illustrates the basic binocular geometry.

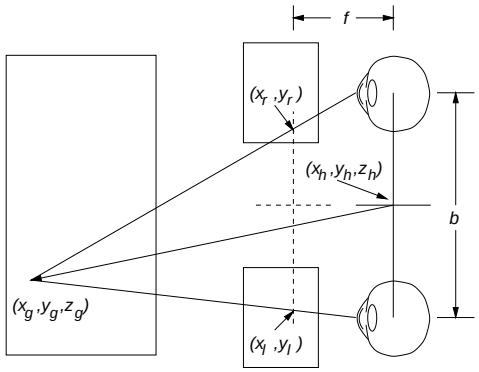


Figure 4: Basic binocular geometry.

Helmet tracking determines both helmet position and the (orthogonal) directional and up vectors, which determine head-centric coordinates. Given instantaneous eye tracked coordinates, (x_l, y_l) and (x_r, y_r), in the left and right image planes (mapped from eye tracker screen coordinates to the near view plane), and head-tracked head position coordinates, (x_h, y_h, z_h), the coordinates of the gaze point, (x_g, y_g, z_g), are determined by the relations:

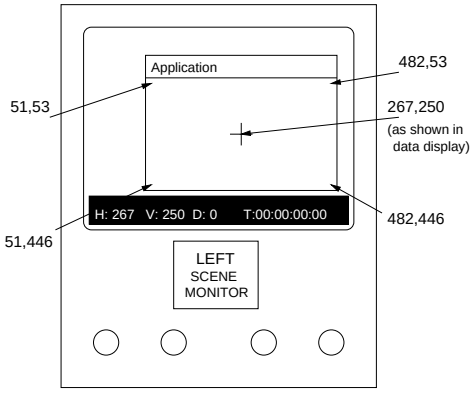
$$x_g = (1-s)x_h + s(x_l + x_r)/2 \quad (1)$$

$$y_g = (1-s)y_h + s(y_l + y_r)/2 \quad (2)$$

$$z_g = (1-s)z_h + sf \quad (3)$$

where $s = b/(x_l - x_r + b)$, b is the baseline distance between the left and right eye centers, and f is the distance to the near viewing plane along the head-centric z-axis.

(a) User wearing eye tracking HMD.



(b) Mapping measurement example.

Figure 3: Eye tracking VR use and measurement mapping.

Note that since the vertical eye tracked coordinates y_l and y_r are expected to be equal (since gaze coordinates are assumed to be epipolar), the vertical coordinate of the central view vector defined by $(y_l + y_r)/2$ is somewhat extraneous; either y_l or y_r would do for the calculation of the gaze vector. However, since eye tracker data is also expected to be noisy, this averaging of the vertical coordinates enforces the epipolar assumption.

To enable collection of fixated points in the environment, it is necessary to calculate the intersection of the user's gaze with the environmental polygons. To calculate gaze direction, the gaze point is expressed parametrically as a point on a ray with origin (x_h, y_h, z_h) with the ray emanating along a vector scaled by parameter s. That is, rewriting Equations (1)–(3) in vector notation, $\mathbf{g} = \mathbf{h} + s\mathbf{v}$, where $\mathbf{h}$ is the head position, $\mathbf{v}$ is the central gaze vector and s is the scale parameter as defined previously. To align the gaze vector with the current head orientation, it is first transformed to the instantaneous head-centric reference frame by multiplying the gaze vector $\mathbf{v}$ by the orientation matrix returned by the head tracker.

The computed gaze direction vector $\mathbf{v}$ is used for calculating gaze/polygon intersections via traditional ray/polygon intersection calculations commonly used in ray tracing [7]. These points, termed here as Gaze Intersection Points (GIPs) for brevity, are each found on the closest polygon to the viewer intersecting the gaze ray, assuming all polygons are opaque. The resulting ray casting algorithm generates a scanpath constrained to lie on polygonal regions within the virtual environment.

6. EYE MOVEMENT ANALYSIS

In general diagnostic eye tracking applications, the purpose of eye movement recording is to catalog the user's (overt) visual attention within the environment over time. A record of the user's fixation sequence (scanpath) can be used to examine attentional qualities of the environment, the user's visual search strategies, or other related cognitive processes. The scanpath usually requires analysis (in real-time or off-line) to distinguish fixations. Current 2D techniques (as explained in the next section) are not always suitable for analysis of eye movements in VR since they tacitly assume that the head is fixed and the line of sight is perpendicular to the view plane.

6.1 Traditional (2D) Approach

The common goal of eye movement analysis techniques is the location of fixations in the eye movement signal over the given stimulus or within stimulus Regions Of Interest (ROIs). Most techniques rely on the measurement of visual angle, where it is often tacitly assumed the head is located at a fixed distance to, and usually also perpendicular to, the stimulus screen. Applicable signal analysis techniques can be grouped into three broad categories: position-variance, velocity-based, and ROI-based. Salvucci and Goldberg provide a good classification of current techniques, which have not changed much since Anliker's classification of the two former methods [15, 1].

In position-variance schemes, the visual angle is used to threshold the stationary portion of the signal (e.g., in terms of position). For example, if gaze remains invariant in an area subtending 5° visual angle for 300ms, then this portion of the signal is deemed a fixation.

In velocity-based schemes, the speed of successive data points is used to distinguish fixations from saccades (the fast, often ballistic, eye movements used to reposition the fovea). This is usually accomplished by thresholding eye movement velocity, expressed in degrees visual angle per second. Anywhere the signal exhibits fast velocity, this portion of the signal is deemed a saccade, and conversely, everywhere else, where velocity is below threshold, the signal can be considered a fixation (or some other type of relatively slow eye movement such as smooth pursuit). The velocity-based saccade detection method can therefore be used as a type of elimination scheme to find fixations in the eye movement signal.

The traditional two-dimensional eye movement analysis approach starts by measuring the visual angle of the object under inspection between a pair (or more) of raw eye movement data points in the time series (i.e., the POR data denoted by (x_i, y_i)). Given the distance between successive POR data points, $r = \| (x_i, y_i), (x_j, y_j) \|$, the visual angle, θ, is calculated by the equation: $\theta = 2\tan^{-1}(r/2D)$ where D is the (perpendicular) distance from the eyes to the viewing plane.

Note that r and D, expressed in like units (e.g., pixels or inches), are dependent on the resolution of the screen on which the POR data was recorded. A conversion factor is usually required to convert one measure to the other.

The visual angle, θ, and the difference in timestamps, Δt, between the POR data points allows velocity-based analysis, since $\theta/\Delta t$ gives eye movement velocity in degrees visual angle per second. A common threshold of 600 deg/s (peak velocity) is used to identify saccades (and hence fixations) [18].

Note that the arctangent approach assumes that D is measured along the line of sight, which is assumed to be perpendicular to the viewing plane. The traditional 2D eye movement analysis can therefore be applied directly to raw POR data in the eye tracker reference frame. As a result, identified fixations could then be mapped to world coordinates to locate ROIs within the environment. We choose a different approach by mapping raw POR data to world coordinates first, followed by eye movement analysis in three-space. We favor this approach since the calculated gaze points in three-space provide a composite three-dimensional representation of both the user's binocular eye movements. Applying the traditional 2D approach suggests a component-wise analysis of left and right eye movements (in the eye tracker's reference frame) effectively ignoring depth, and in so doing possibly discounting the effects of vergence eye movements. In three dimensions, vergence eye movements are implicitly taken into account prior to analysis. However, the assumption of a perpendicular visual target plane does not hold since the head is free to translate and rotate within 6 degrees of freedom.

6.2 New (3D) Analysis Algorithm

In the few VR eye tracking studies currently being conducted, fixation analysis is often not well documented, or restricted to eye-in-head measurements. For example, Tanriverdi and Jacob's work on visual selection in an interactive VR system is based on accumulation of fixations on potential target objects, however, the details of the fixation algorithm are not provided [17]. This may be due to the use of a proprietary line of sight algorithm[2] or due to the interactive nature of the system, where only detection of instantaneous fixations is relevant. The algorithm used by Tanriverdi and Jacob most likely operates on eye-in-head measurements and is effectively an extension of the traditional Region Of Interest (ROI)-based approach (e.g., fixations are classified based on location of gaze and dwell time).

Here, we present a velocity-based algorithm that operates directly on GIP data in (virtual) world coordinates. Given raw gaze intersection points in three dimensions, the velocity-based thresholding calculation is in principle identical to the traditional 2D approach, with the following important distinctions:

1. The head position, $\mathbf{h}$, must be recorded to facilitate the calculation of the visual angle.
2. Given two successive GIP data points in three-space, $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and the head position at each instance, $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$, the visual angle θ is calculated from the dot product of the two gaze vectors defined by the difference of the gaze intersection points and averaged head position:

$$\theta = \cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|} \tag{4}$$

   where $\mathbf{v}_i = \mathbf{p}_i - \overline{\mathbf{h}}$ and $\overline{\mathbf{h}}$ is the averaged head position over the sample time period. Head position is averaged since the eyes can accelerate to reach a target fixation point much more quickly than the head [20].

With visual angle, θ, and timestamp difference between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, the same velocity-based thresholding can be used as in the traditional 2D case. Because all calculations are performed in world coordinates, no conversion between screen resolution and distance to target is necessary.

[2]Tanriverdi and Jacob report the use of ISCAN's Headhunter Line of Sight Computation and Plane Intersection Algorithm, v1.0

The current eye movement analysis algorithm (detection of fixations) effectively relies on the use of a short 2-tap temporal differential filter to estimate velocity. That is, using Equation (4) to calculate θ, only two successive data points are used to calculate eye movement velocity. This is analogous to the calculation of velocity using a Haar convolution filter (e.g., $\{-1/\sqrt{2}, 1/\sqrt{2}\}$ in 1D), for measuring the spatial distance between points. The use of such a short filter is known to be inferior to fixation algorithms based on positional-variance characteristics of the eye movement signal [15]. However, due to its simplicity, the algorithm can be implemented in real-time, making it suitable for an interactive eye tracking VR application. Furthermore, by changing the subscript $i+1$ to $i+k$ for $k > 1$, the algorithm generalizes to the use of wider filters for improved smoothing.

7. DEVICE / SOFTWARE CALIBRATION

In practice, determination of the scalar s (dependent on inter-pupillary distance, or baseline, b) and focal distance f used in Equations (1)–(3) is difficult. Inter-pupillary distance cannot easily be measured in VR since the left and right eye tracking components function independently. That is, there is no common reference point.

During preliminary experiments, calculated GIPs were compared against raw POR video footage. Frame-by-frame visual inspection of video recordings revealed a discrepancy between calculated GIPs and the visual features subjects appeared to be fixating. Since this error appeared to be variable between but consistent within subjects and thought to be related to the unknown inter-pupillary distance, a 3D calibration procedure was designed to estimate the inter-pupillary distance scaling factor s empirically.

The 3D calibration relies on a specially marked environment, containing 9 clearly visible fixation targets, illustrated in Figure 7. (This figure is reproduced in color on page 193.) The 9 $\times$ targets are distributed on 5 walls of the environment to allow head movement to be taken into account during analysis. Without a precise estimate of b and f, computed GIPs may appear stretched or compressed in the horizontal or vertical direction, as shown in Figure 7(a) (only 5 targets are visible in the figure).

Prior to each experimental trial, the user must first complete two short calibration trials: (1) a 5-point 2D calibration sequence to calibrate the eye tracker itself, and (2) the new 3D calibration to enable accurate GIP calculation. To shorten the trial duration, eye movement data is stored for off-line analysis. The scalar parameter s is obtained manually through the use of a simple interface, shown in Figure 5. As the operator manipulates the scale factor sliders, GIP data is re-calculated and displayed interactively. The goal is to align the calculated GIP locations with the environmental targets which the user was instructed to fixate during calibration. An example of this type of adjustment is shown in Figure 7(b). Notice that the GIPs (represented by green, transparent spheres) are now better aligned over the red targets than the raw data in Figure 7(a). Once determined, the scale factor s is used to adjust each participant's eye movement data in all subsequent trials.

8. EXPERIMENTAL VALIDATION

An experiment was conducted to measure the training effects of the VR aircraft inspection simulator. The objectives of the experiment included: (1) validation of performance measures used to gauge training effects, and (2) evaluation of the eye movement data as cognitive feedback for training. Assuming eye movement analysis correctly identifies fixations and the VR simulator is effective for training (i.e., a positive training effect can be measured),

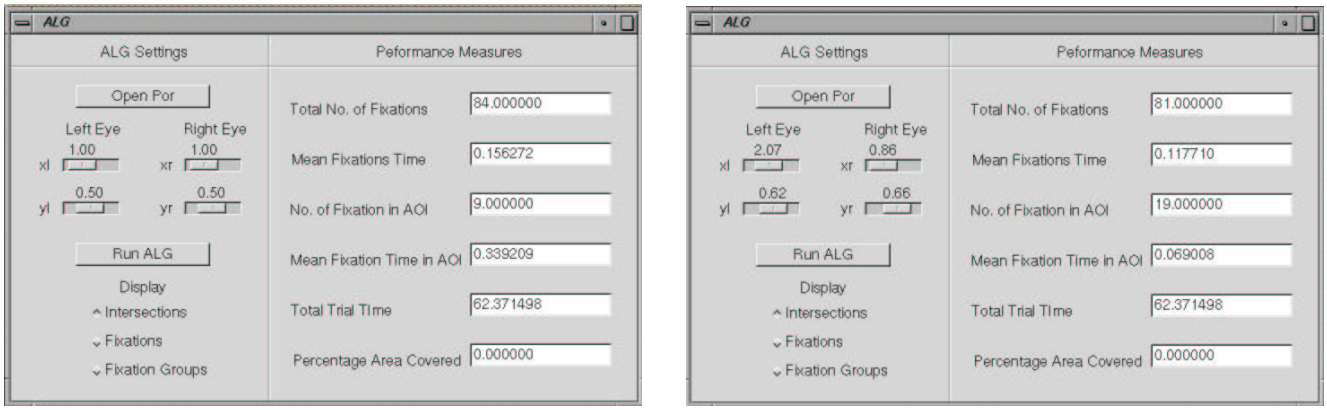


Figure 5: Prior to (a, left), and following (b, right) scaling left and right eye movement data.

the number of detected fixations are expected to decrease with the adoption of an improved visual search strategy [3] (e.g., following training).

8.1 Stimulus Environment

The airframe inspection simulation featured inspection of an aircraft cargo bay with dimension similar to that of a real cargo bay of an L1011 aircraft. Texture maps used in the virtual aircraft cargo bay were created from photographs of an actual cargo bay (see Section 3).

For user interaction with the virtual environment, and performance measurement during immersion, a 6DOF mouse was used as a multi-modal device (see Section 4). The 6DOF mouse allows subjects to perform a pointing and clicking function to indicate selection. The criterion task consisted of inspecting the simulated aircraft cargo bay in search of defects. Several defects can occur in a real environment situation. Three types of defects were selected to create inspection scenarios:

1. Corrosion: represented by a collection of gray and white globules on the inner walls of the aircraft cargo bay and located roughly at knee level.
2. Cracks: represented by a cut in any direction on the structural frames inside the aircraft cargo bay.
3. Damaged conduits: shown as either broken or delaminated electrical conduits in the aircraft cargo bay.

Figure 8(a) shows an example of corrosion defects. As shown in Figure 8(b), target defects are highlighted for the operator but are not typically displayed for the subject. (This figure is reproduced in color on page 193.)

8.2 Performance and Process Measures

Data for performance and cognitive feedback measures was collected using search timing and eye movement information, respectively. The following performance measures were collected:

1. Search time from region presentation to fault detection.
2. Incremental stop time when subjects terminated the search in a region by deciding the region does not contain faults.
3. Number of faults detected (hits), recorded separately for each fault type.
4. Number of faults that were not identified (misses).

Fixation analysis enabled the collection of cognitive feedback measures, which were provided to subjects during the training session. Cognitive feedback measures were based on the eye movement parameters that contribute to search strategies as defined by Megaw and Richardson [13], including: (1) total number of fixations; (2) mean fixation duration; (3) percentage area covered; and (4) total trial time. Cognitive feedback measures were graphically displayed off-line by rendering a 3D environment identical to the aircraft cargo bay which was used during immersive trials. This display represented the scanpaths of each trial to indicate the subject's visual search progression.

8.3 Subjects

Eighteen graduate students were chosen as subjects, all in the 20-25 year old age group. Subjects were screened for 20/20 corrected vision. Subjects were randomly assigned to three different groups (6 per group): Performance Feedback Group (PFG), Cognitive Feedback Group (CFG), and Cognitive + Performance Feedback Group (CPFG). Subjects received different forms of feedback during training sessions before and after trials (see below).

8.4 Experimental Design

The study used a 3×2 experimental design with 3 groups (PFG, CFG, and PCFG) and 2 trials (before training and after training). Six subjects were placed in each of the three groups. Grouping allowed testing of between-subject factors, while within-subject factors were tested between trials. Performance and cognitive feedback measures together constitute 8 dependent variables, with training scenarios (immersion in different defect inspection scenarios) serving as the independent variable (training treatment).

8.5 Procedure

Each subject was requested to complete a consent form and demographic questionnaire. Written and oral instructions were provided to ensure subjects' understanding of the experiment. All subjects were given information about their required task. Subjects were then shown the entire search area of the virtual aircraft cargo bay and were provided with graphical and verbal descriptions of possible types of defects. Subjects were then presented with a familiarization task similar to the actual trials in the Virtual Reality simulator and were shown how to use the 6DOF mouse for pointing at and selecting targets.

The before training criterion task was an unpaced visual inspection search task. Subjects searched for defects on the walls, floor, and the ceiling of the simulated 3D cargo bay. The entire search task was divided into a series of six subtasks listed in Table 1. To cancel out order effects, all six participants in each group completed their assigned subtasks following a counterbalanced order using a 6×6 Latin square design. Treatments were randomly assigned to each of the six participants.

Table 1: Description of subtasks.

#	Scenario	Task Description
1	No (zero) defect	Search entire area with no defects
2	Single defect	Find corrosion defects
3	Single defect	Find crack defects
4	Single defect	Find damaged conduit defects
5	Multiple defect	Find all three defects
6	No (zero) defect	Search entire area with no defects

On completion of the before training trials, all subjects underwent respective training sessions for each of the three groups. The first step in the training sessions was completion of a multi-defect search task. Subjects received feedback training according to the respective feedback training groups:

- Performance Feedback Group. Subjects in this group received performance measures feedback performance (search times, errors).
- Cognitive Feedback Group. Subjects in this group received two forms of cognitive feedback: statistical and graphical. Statistical feedback included the number of fixations, mean fixation duration, number of fixations in ROIs, mean fixation duration in the ROIs, and percentage area covered. For graphical feedback, subjects viewed a graphical visualization of their scanpaths representing their search patterns with fixation indices showing their visual search progression.
- Cognitive + Performance Feedback Group. Subjects in this group received both forms of feedback, performance feedback training as well as cognitive feedback training.

On completion of the training sessions, all subjects performed an after training criterion task. This subtask was counterbalanced to eliminate order effects.

9. RESULTS

Analysis of variance (ANOVA) showed no significant differences between subjects (feedback groups). However, ANOVA showed significant differences in mean search time, percentage defects detected, incremental stopping time, and total trial time within subjects.

9.1 Eye Movement Analysis

Assuming typical fixation durations of range 150ms–600ms and 90% of search time spent in fixations [11], we expected to find observed fixation statistics to be similarly distributed. The overall measured mean fixation duration was 112ms over 12 total trials for all 18 subjects. We recorded an average of 236 fixations during each trial. Based on total measured trial time of each subject, we expected the number of fixations to range between 265 and 1345 fixations.

Our automatic 3D fixation algorithm underestimates fixation duration and also underestimates the number of fixations. This result was not wholly unexpected. The velocity-based saccade detection method is known to be a weak fixation detector when used in isolation. However, it is often a necessary first step to locating slow-moving eye movements which can then be processed further to isolate and group fixation points. Furthermore, as expected, we noted a high degree of noise in the data. The two main sources of noise are most likely the eye tracker and the short filter used in the velocity-based algorithm.

The eye tracker is inherently somewhat noisy, and frequently delivers null POR values, usually coinciding with blinks. Sample data with null values for either the left or right POR is automatically eliminated by our algorithm. Over all trials, we observed an estimated mean 10% data loss. Considering mean trial durations of 177s and a sample rate of 60Hz, this data loss rate is quite high. Unfortunately, little can be done to improve eye tracker performance.

The short filter used in the velocity-based analysis is another source of noise. The filter is mathematically appropriate for calculating velocity, but due to its short length, it is known to be quite noisy. We believe, however, that the filter gives a good first approximation, and because of its short length, is suitable for real-time applications. For more robust off-line fixation analysis either a longer filter should be used or a hybrid signal analysis approach should be considered (e.g., an approach where position-variance analysis is applied to the results of the velocity-based algorithm to accumulate fixations).

The current eye movement analysis algorithm suffers from a rather simplistic signal processing approach incapable at this point of robustly dealing with a noisy signal. Nevertheless, the analysis algorithm is applied consistently over the duration of the signal, and does not appear to introduce variance in results before and after experimental treatments. Although excessively sensitive to noise, our eye movement analysis still permits us to evaluate its utility in terms of training effects and process measures.

9.2 Process Measures & Training Effects

Analysis indicates that, overall, training in the VR aircraft simulation has a positive effect on subsequent search performance in VR, although there is apparently no difference in the type of feedback given to subjects. Cognitive feedback, in the form of visualized scanpaths, does not appear to be any more effective than performance feedback. It may be that the common most effective contributor to training is the immersion in the VR environment, that is, the exposure to the given task, or at least to the simulated task.

Whether the eye tracker, by providing cognitive feedback, contributes to the improvement of inspection performance is inconclusive. Users may benefit just as much from performance feedback alone. However, the eye tracker is a valuable tool for collecting process measures. Analysis of results leads to two observations. First, mean fixation times do not appear to change significantly following training. This is not surprising since eye movements are to a large extent driven by physiology (i.e., muscular and neurological functions). Second, the number of fixations decrease following training. While at this time analysis prevents us from reporting statistical significance (we believe due to the observed high rate of data loss), we can descriptively report an apparent trend in the reduction of the number of fixations observed in post-training trials. Within-group percentage reduction of the number of fixations is given in Table 2. These results generally appear to agree with

Table 2: Within group fixation count reduction.

Group	% reduction
Cognitive Feedback	5.42
Performance Feedback	18.94
Cognitive + Performance Feedback	2.85

the expectation of reduced number of fixations with the adoption of an improved visual search strategy (e.g., due to learning or familiarization of the task). The implication of reduced number of fixations (without an increase in mean fixation time) suggests that in the post-training case, subjects tend to employ a greater number of saccadic eye movements. That is, an improved visual search strategy may be one where subjects inspect the environment more quickly (perhaps due to familiarity gained through training), reducing the time required to visually rest on particular features.

In summary, performance measures quantify the level of improvement of subjects' inspection performance (i.e., *how* the subject performed). If improvement can be shown, then we may conclude that training contributes to performance improvement and additionally that the VR simulator is a suitable environment for training. Process measures can not only corroborate performance gains, but can also lead to discoveries of reasons for performance improvements (i.e., *what* the subject performed). In particular, tracking the users' eyes can potentially lead to further insights into the underlying cognitive processes of human inspectors.

10. CONCLUSION

This paper presented novel software techniques developed for binocular eye tracking within Virtual Reality for aircraft inspection training. Methods were given for (1) integration of the eye tracker into a Virtual Reality framework, (2) stereo calculation of the user's 3D gaze vector, (3) a new 3D calibration technique developed to estimate the user's inter-pupillary distance *post-facto*, and (4) a new technique for eye movement analysis in 3-space. The new 3D eye movement analysis technique is an improvement over traditional 2D approaches since it takes into account the 6 degrees of freedom of head movements and is resolution independent. Results indicate that although the current signal analysis approach is somewhat noisy and tends to underestimate the identified number of fixations, recorded eye movements provide valuable human factors process measures complementing performance statistics used to gauge training effectiveness.

11. ACKNOWLEDGMENTS

This work was supported in part by a University Innovation grant (# 1-20-1906-51-4087) and NASA Ames task (# NCC 2-1114).

12. REFERENCES

[1] ANLIKER, J. Eye Movements: On-Line Measurement, Analysis, and Control. In *Eye Movements and Psychological Processes*, R. A. Monty and J. W. Senders, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 1976, pp. 185–202.

[2] DANFORTH, R., DUCHOWSKI, A., GEIST, R., AND MCALILEY, E. A Platform for Gaze-Contingent Virtual Environments. In *Smart Graphics (Papers from the 2000 AAAI Spring Symposium, Technical Report SS-00-04)* (Menlo Park, CA, 2000), AAAI, pp. 66–70.

[3] DRURY, C. G., GRAMOPADHYE, A. K., AND SHARIT, J. Feedback Strategies for Visual Inspection in Airframe Structural Inspection. *International Journal of Industrial Ergonomics 19* (1997), 333–344.

[4] DUCHOWSKI, A., KARN, K. S., AND SENDERS, J. W., Eds. *Eye Tracking Research & Applications (ETRA)* (Palm Beach Gardens, FL, 2000), ACM. URL: <http://www.vr.clemson.edu/eyetracking/et-conf/>, last accessed 9/7/00.

[5] DUCHOWSKI, A. T. Incorporating the Viewer's Point-Of-Regard (POR) in Gaze-Contingent Virtual Environments. In *Stereoscopic Displays and Virtual Reality Systems V* (Bellingham, WA, January 25-January 30 1998), SPIE.

[6] DUCHOWSKI, A. T., AND VERTEGAAL, R. *Course 05: Eye-Based Interaction in Graphical Systems: Theory & Practice*. ACM SIGGRAPH, New York, NY, July 2000. SIGGRAPH 2000 Course Notes, URL: <http://www.vr.clemson.edu/eyetracking/sigcourse/>, last accessed 9/7/00.

[7] GLASSNER, A. S., Ed. *An Introduction to Ray Tracing*. Academic Press, San Diego, CA, 1989.

[8] GRAMOPADHYE, A., BHAGWAT, S., KIMBLER, D., AND GREENSTEIN, J. The Use of Advanced Technology for Visual Inspection Training. *Applied Ergonomics 29*, 5 (1998), 361–375.

[9] HELD, R., AND DURLACH, N. Telepresence, time delay and adaptation. In *Pictorial Communication in Virtual and Real Environments*, S. R. Ellis, M. Kaiser, and A. J. Grunwald, Eds. Taylor & Francis, Ltd., London, 1993, pp. 232–246.

[10] HORN, B. K. P. *Robot Vision*. The MIT Press, Cambridge, MA, 1986.

[11] IRWIN, D. E. Visual Memory Within and Across Fixations. In *Eye Movements and Visual Cognition: Scene Perception and Reading*, K. Rayner, Ed. Springer-Verlag, New York, NY, 1992, pp. 146–165. Springer Series in Neuropsychology.

[12] JACOB, R. J. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings* (1990), ACM Press, pp. 11–18.

[13] MEGAW, E. D., AND RICHARDSON, J. Eye movements and industrial inspection. *Applied Ergonomics 10*, 3 (1979), 145–154.

[14] OHSHIMA, T., YAMAMOTO, H., AND TAMURA, H. Gaze-Directed Adaptive Rendering for Interacting with Virtual Space. In *Proceedings of VRAIS'96* (March 30–April 3 1996), IEEE, pp. 103–110.

[15] SALVUCCI, D. D., AND GOLDBERG, J. H. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Eye Tracking Research & Applications (ETRA) Symposium* (Palm Beach Gardens, FL, 2000), ACM, pp. 71–78.

[16] STARKER, I., AND BOLT, R. A. A Gaze-Responsive Self-Disclosing Display. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings* (1990), ACM Press, pp. 3–9.

[17] TANRIVERDI, V., AND JACOB, R. J. K. Interacting with Eye Movements in Virtual Environments. In *Human Factors in Computing Systems: CHI 2000 Conference Proceedings* (2000), ACM Press, pp. 265–272.

[18] TOLE, J. R., AND YOUNG, L. R. Digital Filters for Saccade and Fixation Detection. In *Eye Movements: Cognition and Visual Perception*, D. F. Fisher, R. A. Monty, and J. W. Senders, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981, pp. 7–17.

[19] VORA, J., NAIR, S., MEDLIN, E., GRAMOPADHYE, A., DUCHOWSKI, A. T., AND MELLOY, B. Using Virtual Technology to Improve Aircraft Inspection Performance: Presence and Performance Measurement Studies. In *Proceedings of the Human Factors and Ergonomics Society* (2001). To appear.

[20] WATSON, B., WALKER, N., AND HODGES, L. F. Managing Level of Detail through Head-Tracked Peripheral Degradation: A Model and Resulting Design Principles. In *Virtual Reality Software & Technology: Proceedings of the VRST'97* (1997), ACM, pp. 59–63.

[21] WITMER, B. G., AND SINGER, M. J. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence 7*, 3 (June 1998), 225–240.