Cognitive Feedback Training using 3D Binocular Eye Tracker

Santosh N. Nair, Anand K. Gramopadhye, Jeenal Vora, Brian J. Melloy
Department of Industrial Engineering, Clemson University

Eric Medlin, Andrew T. Duchowski,
Department of Computer Science, Clemson University

Barbara Kanki
Human Factors Research and Technology Division
NASA Ames Research Center

Studies in visual search have shown that feedback training can improve visual inspection performance (speed and accuracy), provided it is given in a timely and appropriate manner. Traditionally, performance feedback, i.e. information about the outcome, serves as the basis of most feedback training schemes. Other forms of feedback, which provide search strategy information, may have a role to play in improving inspection performance. This form of feedback is referred to as ' cognitive feedback' . This paper describes the setup for collecting and analyzing eye movements using a 3D binocular eye tracker, which serves as a tool for providing cognitive feedback training. It also emphasizes the various technical issues related with eye tracker integration and further discusses the use of a new 3D-fixation recognition algorithm.

INTRODUCTION

Aircraft inspection and maintenance are an essential part of a safe, reliable air transportation system. Visual inspection is important as it accounts for almost 90 % of aircraft inspection (Drury et al., 1990). Since the time spent by the aircraft in maintenance represents a large loss of revenue, the inspection system must improve in its effectiveness and efficiency. Training has been identified as the primary intervention strategy in improving inspection performance (Gramopadhye et al, 1998). If training is to be successful, it is clear that inspectors need to be provided with training tools to help enhance their inspection skills.

There have been several training strategies documented in the literature for visual inspection (Gramopadhye, Drury and Prabhu, 1997), one among them is feedback training. Traditionally, the feedback provided to the inspectors has been performance feedback, which consists of feedback on search times, search errors (faults not detected) and decision errors (Micalizzi and Goldberg, 1989). Investigators have reported improved performance by providing performance feedback (Gramopadhye et al, 1996). However, it is possible to give process feedback, which provides the trainee with task information on the search processes and strategies. This kind of feedback is referred to as ' cognitive feedback' (Gramopadhye et al., 1996).

Applications of cognitive feedback in the visual inspection domain have been very rare. Its application in non-inspection situations such as learning tasks (Lindell, 1976) has been shown to have beneficial effects. In a study conducted by Deane (1972), cognitive feedback was found to result in a better understanding of the task characteristics and improved control performance by humans over the execution of their knowledge. In a more recent review of decision studies (Balzer et al., 1989), relating to both realistic and abstract tasks, cognitive feedback showed consistent success in providing individuals with an insight into their own policies and strategies. Search performance in a visual inspection task depends on the inspector' s ability to understand the task characteristics and his decision to adopt an efficient search strategy. Providing cognitive feedback can prove to be beneficial.

Cognitive feedback consists primarily of three feedback modes in which information can be given to the inspectors. They are auditory cognitive feedback, statistical cognitive feedback and graphical cognitive feedback. For visual search, auditory cognitive feedback would seem inappropriate, as input is primarily visual and spatial, with output typically being a motor action, whereas statistical cognitive feedback and graphical cognitive feedback would seem to be more appropriate. In the past, visual inspection literature has neither provided information on a better form of cognitive feedback, nor a comparison of cognitive feedback to traditional performance feedback. The only study in the visual inspection literature, that addressed this issue, was by Gramopadhye, Drury and Sharit (1996). Their study showed that the effect of cognitive feedback alone did not affect search performance (speed and accuracy) but it did significantly influence the search strategy. In their study a movable window, i.e. the field of view or ' viewer' as it was explained to the subjects, could be moved around the inspection area using the mouse, exposing whatever was within the viewer' s field of view. This viewer was used as tool for collecting cognitive feedback information and was analogous to collecting eye movement data. The authors' rationalized the use of the viewer as a tool for collecting eye movement data, based on cost and ease of data collection. However, this method is intrusive, relies only on foveal vision and may not reflect true search behavior. Moreover, their

results on the use of cognitive feedback may have been an artifact of the experimental setup.

In recent years, reduced cost, improved accuracy and high computational capability have enabled researchers to use non-intrusive eye trackers for a range of visual applications. This method is more accurate and is reflective of actual search behavior. Examples of this can be found in the studies conducted by Goldberg and Kotval (1998) and Wang et al (1997). These studies reported successes in understanding visual search strategies for graphical interface design and training applications respectively. Drawing from the results of these studies and the earlier study by Gramopadhye et al (1996), we see that eye movement recordings as a cognitive feedback tool may have a role to play in improving visual search performance. This paper describes the setup for collecting and analyzing eye movements in a virtual reality inspection based environment integrated with a 3D binocular eye tracking system.

EYE TRACKING IN VIRTUAL REALITY ENVIRONMENT

Interest in gaze-contingent 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 (Jacob, 1990; Starker and Bolt, 1990; Held and Durlach, 1993). Recent applications of an eye tracker in virtual reality environment have shown promising results in the use of the device as a component for multi-modal interface systems. Some of the examples are Jacob and Tanriverdi (2000) who used an eye tracker as a selection device in virtual reality environment and Danforth et al., (2000) who used an eye tracker as an indicator of gaze in a gaze-contingent, multi-resolution terrain navigation environment. There are two types of applications of an eye tracker: one is to use it as an interactive device and the other is to serve as a diagnostic tool. The eye tracking system discussed in this paper serves as a diagnostic tool with the user's eye movements unobtrusively being recorded in real time for post-immersion analysis.

Eye Tracker Coordinate Mapping

The eye movement data obtained from the tracker must be mapped to a range appropriate for the VR application. Specifically, the 2D eye tracker data, expressed in eye tracker screen coordinates, must be mapped to the 2D dimensions of the near viewing frustum. The parameters left, right, bottom, top, near and far, defines the 3D viewing frustum employed in the perspective viewing transformation. Figure 1 shows the dimensions of the eye tracker screen (left) and the dimensions of the viewing frustum (right).

To convert the eye tracker coordinates (x', y') to graphics coordinates a linear interpolation mapping is used:

$$x = \texttt{left} + \frac{x'(\texttt{right} - \texttt{left})}{512} \quad (1)$$

$$y = \texttt{bottom} + \frac{(512 - y')(\texttt{top} - \texttt{bottom})}{512} \quad (2)$$

Since the eye tracker, origin is at the top-left of the screen and the viewing frustum's origin is at the bottom-left (a common discrepancy between imaging and graphics applications), the term $(512 - y')$ in Equation (2) handles the necessary y-coordinate mirror transformation.

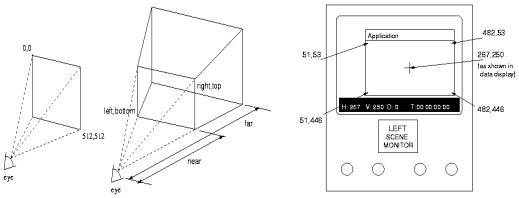


Figure 1. Coordinate mapping of eye tracker to 3D viewing frustum screen and example of mapping measurement.

The above coordinate mapping assumes that the eye tracker coordinates are in the range $[0,511]$. In reality, the usable, or effective, coordinates will be dependent on (i) the size of the application window, and (ii) the position of the application window. 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 own 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. These window extents are then used in the linear mapping equation. Figure 1 illustrates an example of a 600×450-application window, as it would appear on the eye tracker scene monitor. Based on the measurements shown in Figure 1, the linear coordinate mapping is:

$$x = \frac{x' - 51}{(482 - 51 + 1)}(600) \quad (3)$$

$$y = 449 - \frac{y' - 53}{(449 - 53 + 1)}(450) \quad (4)$$

The central point on the eye tracker display is (267,250). Note that y is subtracted from 449 to take care of the image/graphics vertical origin flip.

Gaze Vector Calculations

The calculation of the point of regard in three-space depends on only the relative positions of the two eyes in the horizontal axis. The parameters of interest here are the three-dimensional virtual coordinates, (x_g, y_g, z_g), which can be determined from traditional stereo geometry calculations.

Figure 2 illustrates the basic binocular geometry. Helmet tracking determines both helmet position and the (orthogonal) directional and up vectors, which determine viewer-local coordinates shown in the diagram. The helmet

position is the origin, the helmet directional vector is the optical (viewer-local z) axis, and the helmet up vector is the viewer-local y-axis.

Given instantaneous, eye tracked, viewer-local 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 viewer-local coordinates of the gaze point, (x_g, y_g, z_g), are determined by the relations:

$$x_g = (1 - s)x_h + s(x_l + x_r)/2 \quad (5)$$

$$y_g = (1 - s)y_h + s(y_l + y_r)/2 \quad (6)$$

$$z_g = (1 - s)z_h + sf \quad (7)$$

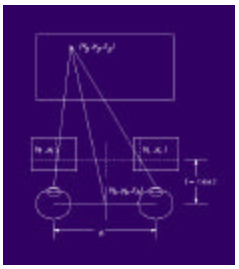


Figure 2. Basic binocular geometry

Where $s = b/(x_l - x_r + b)$, b is the disparity distance between the left and right eye centers, and f is the distance to the near viewing plane along the viewer-local z-axis. Gaze point coordinates based on mergence calculations given by Equations (5) --- (7) are presently closely correlated with the user's head location. The right image shows the collection of gaze points from a side viewpoint. With respect to depth, the gaze points do not precisely fall on the polygonal surfaces of the environment.

To calculate the gaze/polygon intersection, the gaze point is expressed parametrically as a point on a ray with origin (x_h, y_h, z_h), the helmet position, with the ray emanating along a vector scaled by parameter s. That is, rewriting Equations (5- 7), we have:

$$x_g = x_h + s\left(\frac{x_l + x_r}{2} - x_h\right)$$

$$y_g = y_h + s\left(\frac{y_l + y_r}{2} - y_h\right)$$

$$z_g = z_h + s(f - z_h)$$

Or, in vector notation,

$$\mathbf{g} = \mathbf{h} + s\mathbf{v} \quad (8)$$

where **h** is the head position, **v** is the central view vector and s is the scale parameter as defined previously. The view vector **v** is obtained by subtracting the helmet position from the midpoint of the eye tracked x-coordinate and focal distance to the near view plane, i.e.,

$$\mathbf{v} = \begin{bmatrix} (x_l + x_r)/2 \\ (y_l + y_r)/2 \\ f \end{bmatrix} - \begin{bmatrix} x_h \\ y_h \\ z_h \end{bmatrix} = \mathbf{m} - \mathbf{h} \quad (9)$$

where **m** denotes the left and right eye coordinate midpoint. To align the view vector to the current head orientation, the vector **m** must first be transformed to the proper (instantaneous) head orientation. This is done by first normalizing **m** and then multiplying it by the orientation matrix returned by the head tracker.

Given the three-dimensional gaze vector, **v**, specified by Equation (9) gives the coordinates of the gaze point parametrically along a ray originating at the head position (x_h, y_h, z_h). The depth of the three-dimensional gaze point in world coordinates is valid only if $s > 0$.

Gaze Point Calculations

The formulation of the gaze direction given by Equation (9) can be used for testing virtual gaze/polygon intersection coordinates via traditional ray/polygon intersection calculations commonly used in ray tracing (Glassner, 1989). The gaze/polygon intersection point is found on the closest polygon to the viewer intersecting the gaze ray, assuming all polygons are opaque. This polygon is found by testing all polygons in the scene for intersection with the gaze ray. To find the intersection point **g** of the gaze ray with the closest polygon, a new interpolant t is obtained by calculating the gaze ray intersections with all scene polygons. All such intersections are examined for which $t > 0$. The interpolant t is obtained by substituting the gaze ray equation into the polygon's plane equation (in vector notation):

$$t = \frac{-(\mathbf{N} \cdot \mathbf{h} + D)}{\mathbf{N} \cdot \mathbf{v}} \quad (10)$$

where **N** is the negated polygon normal and D is the height parameter of the polygon's plane equation. The geometry of this calculation is depicted in Figure 3.

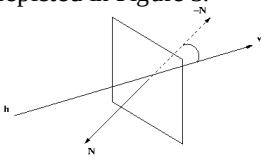


Figure 3. Ray/plane geometry.

The calculation of the ray/plane intersection may be speeded up by evaluating the denominator of Equation (10) first. The intersection algorithm is given below.

```
v_d = N · v;                          // Denominator
if (v_d < 0) {
        v_o = -(N · h + D);           // Numerator
        t = v_o / v_d;
}
```

Note that the ray/polygon intersection algorithm only returns the intersection point of the ray and the infinite plane defined by the polygon's face normal. Because the normal defines a plane of infinite extent, the point **g** must be tested against all of the polygon's edges to establish whether the point lies inside the polygon. This is an instance of a solution to the well-known "point-in-polygon" problem. If the point g is bounded by the perpendicular planes defined by the polygon's edges, then g lies within the polygon, otherwise it lies on the plane defined by the face normal N, but outside the polygonal region. The resulting algorithm generates a scan path constrained to lie on polygonal regions within the virtual environment.

3D Eye Movement Analysis

In diagnostic eye tracking virtual reality (VR) 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 (scan path) can be used to examine attentional qualities of the environment, the user's visual search strategies, or other related cognitive processes. The scan path is usually analyzed (in real-time or off-line) to distinguish fixations. Current 2D techniques 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. 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, Jacob's (2000) work on visual selection in an interactive VR system is based on the accumulation of fixations on potential target objects. The algorithm used by Jacob operates on eye-in-head measurements and is effectively an extension of the traditional region of interest (ROI)-based approach. The algorithm classifies fixations based on location of gaze and dwell time. This paper presents a velocity-based algorithm that operates directly on point of regard (POR) data, mapped to the (virtual) world coordinates.

Traditional Eye Movement Analysis

Traditional eye movement signal analysis techniques can be grouped into three broad categories: position-variance, velocity-based, and ROI-based (Salvucci and Goldberg, 2000). The common goal of these techniques is the location of saccades and fixations in the eye movement signal over the given stimulus (or within stimulus ROIs, as in the latter class of algorithms). 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 perpendicular to, the stimulus screen. The traditional approach starts by measuring the visual angle of the object under inspection on a pair (or more) of raw eye movement data points. Each such data point, usually denoted by (x_i,y_i), is referred to as the Point Of Regard. Given the distance between POR data points, r, 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 (Duchowski et al, 2000). The visual angle, θ, and the difference in time stamps, 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).

New (3D) Eye Movement Analysis

In VR, an eye tracker is used to obtain raw POR data relative to the eye tracker's screen reference frame. The measured POR coordinate pair must be mapped to the extents of the application program's view port. A simple linear interpolation mapping can be used for this purpose. Care must be taken to measure the usable, or effective, application window extents, which depend on (i) the size of the application window, and (ii) the position of the application window relative to the eye tracker's reference frame. In VR, the application must also record the coordinates of the head's position and orientation. This information is usually provided by the head-tracking device, and is used to transform the mapped POR data to world coordinates in the head-centric reference frame.

3 D Fixation Algorithm

Once the real-time POR is appropriately mapped into the world coordinates, the participant's scan path is obtained by calculating points of intersection of the user's gaze with environmental polygons, termed here as gaze intersection points, or GIPs. Given the raw GIPs in three dimensions, the principle of the velocity-based calculation is identical to the traditional 2D approach, with the following important distinctions:

1. The head position, **h**, must be recorded to facilitate the calculation of the visual angle.
2. Given two POR data points in three-space, $P_i = (x_i,y_i,z_i)$ and $P_{i+1} = (x_{i+1}, y_{i+1},z_{i+1})$ 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 POR and head position:

$$q = COS^{-1}\frac{g_i . g_{i+1}}{\| g_i \| \| g_{i+1} \|}$$

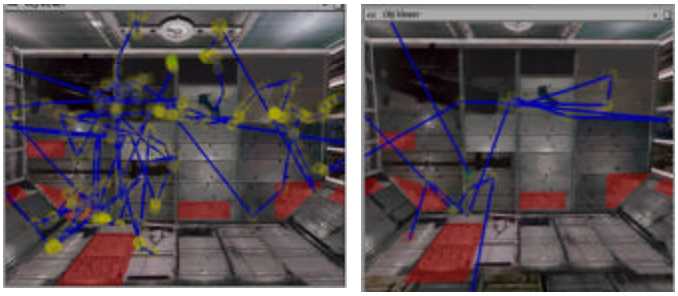
where $g_i = p_i - \bar{h}$ and $\bar{h}$ is the averaged head position the sample time period. With visual angle, θ, and the time stamp difference between p_i and p_{i+1}, the same velocity-based threshold can be used as in the traditional 2D case. Because all calculations are formed in world coordinates, no conversion between screen resolution and distance to target is necessary.

In order to validate the data generated by the fixation algorithm, a search task trial was performed. The eye tracker used is a binocular ISCAN unit built into a Virtual Research V8 Head Mounted Display (HMD). Although the eye tracker operates at 60Hz (video rate), due to software limitations the effective sampling rate was 30Hz. HMD position and orientation tracking was by an Ascension Flock Of Birds (FOB).

Assuming fixation durations of range 150ms–600ms (Irwin, 1992), the expected number of fixations for a trial run of 17.113 seconds in length is approximately 45 (with range of 28–114 fixations of 600ms and 150ms durations, respectively). The 3D-fixation algorithm detected 293 fixations, clearly overestimating the number of fixations. This result is not surprising, however, and is mostly likely due to the short filter used in the velocity-based analysis. The filter is mathematically appropriate for calculating velocity, but due to its short length, it is known to be quite noisy. However, 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, a longer filter should be used. Hence, further evaluation of the algorithm is required.

Tool for cognitive feedback training

Cognitive feedback for a visual inspection task can be provided in two forms: statistical cognitive feedback and graphical cognitive feedback. The data gathered from the system for statistical cognitive feedback consists of total number of fixations identified in the complete trial, mean fixation times, total number of fixations identified in area of interest (AOI, defined as the area around a defect target), mean fixation times in the AOI, total search times and percentage of area covered during the trail. Graphical feedback data is displayed using a 3D-graphical feedback window, which loads an inspection scenario and overlays it with viewer scan paths gathered during the actual run. Figure 9 shows two types of graphical displays, the raw scan path (Figure 9a) and the fixation scan path (Figure 9b). The raw scan path is formed by the point-of–regard points and the fixations scan path is formed by the points identified by the fixation algorithm. Every display identifies the start and the end of the scan path using a color coding scheme which represents the start of the fixation with a green color dot and the end with a red color dot.



(a) Raw Scan path (b) Fixation Scan path

Figure 9. Display of Graphical Cognitive Feedback.

CONCLUSION

This paper outlines the use of 3D-eye movement analysis to provide cognitive feedback using a virtual reality aircraft inspection environment. Furthermore, the paper describes an operational platform for real-time recording of eye movements in a Virtual Reality environment. The platform is based on high-end graphics engines and an electro magnetically tracked, binocular helmet equipped with infrared eye tracking capability. Rendering techniques are relatively simple, relying only on standard (OpenGl) graphics library calls. Tracking routines deliver helmet position and orientation in real-time, which are used directly to provide, updated images to the HMD. User gaze direction can be tracked in real-time, along with calculated gaze/polygon intersections. This process helps in analyzing recorded gaze intersection points for comparison with stored locations of artificially generated defects in the inspection environment. The use of a VR based inspection environment will enable us to conduct controlled studies and address visual search strategy issues for aircraft inspection training.

ACKNOWLEDGEMENT

This research was funded by grants to Dr. Gramopadhye, Dr. Duchowski and Dr. Melloy from NASA AMES Research Center (Program Manager: Barbara Kanki and from the Office of Aviation Medicine, Aviation Maintenance & Inspection: Human Factors Research Program, Federal Aviation Administration (Program manager: Jean Watson).

REFERENCES

1. Balzer, W.K., Doherty, M.E. and O' Conner, R.O., 1989, Effects of cognitive feedback in performance. Psychological Bulletin, 106: 410-433.
2. Czaja, S.J. and Drury, C.G., 1981, Training programs for inspection, Human Factors, 23(4): 473-484.
3. Deane, D.H., Hammond, K.R. and Summers, D.A.., 1972, Acqusition and application of knowledge in complex inference tasks. Journal of Experimental Psychology, 92: 20-26.
4. Drury, C.G. and Addison, J.L., 1973, An industrial study of the effects of feedback and fault density on inspection performance, Ergonomics, 16(2): 707-718.
5. Drury, C. and Gramopadhye, A., 1990, Training for visual search, In *Proceedings of the 3rd FAA Meeting on Human Factors in Aircraft Maintenance and Inspection: Training Issues*. Atlantic City, New Jersey.
6. Duchowski, A.T., Shivashankaraiah, V., Rawls, T., Gramopadhye, A.K., Melloy, B., Kanki, B., 2000, Binocular Eye Tracking in Virtual Reality for Inspection Training. In *Eye Tracking Research &Applications Symposium* (Palm Beach Gardens, FL), ACM.
7. Gramopadhye, A.K., Drury, C.G. and Sharit J., 1996. Feedback strategies for visual search in airframe structural inspection. International Journal of Industrial Ergonomics, 19(1997): 333-344.
8. Gramopadhye, A., Bhagwat, S., Kimbler, D., and Greenstein, J., 1998, The Use of Advanced Technology for Visual Inspection Training. *Applied Ergonomics 29*, 5, pp. 361-375.
9. Gramopadhye, A., Ivaturi, S., Blackmon, R.B., and Kraus, D. C., 1995, Teams and teamwork: Implications for team training within the aircraft inspection and maintenance environment, *FAA-1995 Technical Report*. Office of Aviation Medicine. FAA.
10. Irwin, D.E. Visual Memory Within and Across Fixations. *In Eye Movements and Visual Cognition: Scene Perception and Reading*, K. Rayner, Ed. Springer, New York, NY, 1992, pp. 145-165. Springer Series in Neuropsychology.
11. Jacob, R. J., 1990, What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*, ACM Press, pp. 11-18.
12. Latorella, K., Gramopadhye, A., Prabhu, P., Drury, C., Smith, M., and Shanahan, D., 1992, Computer-simulated aircraft inspection tasks for off-line experimentation, In *Proceedings of the Human Factors Society 36th Annual Meeting*, pp. 92-96.
13. Lindell, M.K., 1976, Cognitive and outcome feedback in multiple cue probability learning tasks. Journal of Experimental Psychology: Human Learning and Memory, 2:739-745.
14. Micalizzi, J. and Goldberg, J., 1989, Knowledge of results in visual inspection decision: sensitivity or criterion effect. *International Journal of Industrial Ergonomics*, 4, pp. 225-235.
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., 1990, A Gaze-Responsive Self-Disclosing Display. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*, ACM Press, pp. 3-9.
17. Tanriverdi, V., and Jacob, R. J. K., 2000, Interacting with Eye Movements in Virtual Environments. In *Human Factors in Computing Systems: CHI 2000 Conference Proceedings*, ACM Press, pp. 265-272.