14. SOFTWARE TUTORIALS/DEMONSTRATIONS

Chaired by George Wolford, *Dartmouth College*

Eye Lines: Generating data through image manipulation, issues in interface design, and the teaching of experimental thinking

WALTER K. BEAGLEY
Alma College, Alma, Michigan

Eye Lines is an experiment generator that differs markedly from the tachistoscope paradigm. Instead of recording reaction times and simple choices, it allows subjects to manipulate images and takes precise measurements of the results. This is useful for experiments in perception, rating scales, or any research needing analog input. A new version further refines a user interface designed to make the programs accessible to beginning students. The interface strategy includes the following: (1) the use of familiar structures, (2) the avoidance of rigid sequences, (3) a new graphic representation for the complexities of reconfiguring data, and (4) the use of context-sensitive defaults and concealment of advanced features. Eye Lines has been used to move beginning students quickly into designing and running their own experiments.

The dominant paradigm in computer research tools for human experimental psychology has been that of the tachistoscope (Abboud, 1991; Eamon, 1982; Enns, Ochs, & Rensink, 1990; Meike, Bharucha, & Baird, 1987; Schneider, 1988). Computers are ideal for presenting stimuli and collecting precisely timed keystrokes. As with any successful paradigm, the usefulness of this approach may be preventing us from seeing other possibilities. Eye Lines was developed to explore one such alternative. It presents stimuli and collects responses, but instead of requiring just simple keystrokes or mouseclicks, Eye Lines allows subjects to manipulate simple images and then records the results. Thus, the primary datum from each trial is not a reaction time or digital choice but rather a precise analog adjustment of some aspect of the image: the position, size, or angular orientation of a line, object, or part of an object. After a brief description of how Eye Lines works, this paper will examine two issues: the problems of designing an interface that presents users with the power of almost unlimited choice while keeping it simple enough for beginners, and how we have used Eye Lines in undergraduate teaching to encourage creativity in experimental design.

Using Eye Lines

The software is useful for any experiment that needs to measure judgments on a continuous scale. Examples include the use of method of adjustment to measure the magnitude of geometric visual illusions, other perception experiments that measure the perceived size, angle, or position of a stimulus, and research in which subjects make judgments on rating scales. To use Eye Lines, one first uses EL Draw to create line drawings, with the options of adding text and linking lines together to form objects. Then, with EL Set, one sets up an experiment by specifying the following for each trial: image name, which elements are to be adjusted, what type of adjustment is to be made, and an optional target value against which to compare the subjects' adjustments. During the experiment (EL Run), the subject uses mouse or cursor keys to manipulate the image. The results are recorded and can later be sorted by EL Sort into a variety of spreadsheet configurations for use with any statistical software package. See Beagley (1991) or Colbourn (1992) for more details.

Issues in Interface Design

If software is to have much impact outside of the lab in which it was created, it must have an interface that is both easy to learn and fun to use (Butler, 1988; Lesgold, 1991). Eye Lines attempts to achieve this by the use of the following strategies.

Familiar structures: Pull-down menus and spreadsheets. Now that mice are affordably available for any

Eye Lines is available from the author for either Macintosh or PC. Prices start at \$99 for a six-copy license. Correspondence should be addressed to W. K. Beagley, Alma College, Alma MI 48801 (e-mail: Beagley@Alma.edu).

machine, the control of programs through mouse-operated pull-down menus is becoming so sufficiently widespread that it requires little or no instruction. In systems such as Eye Lines, which runs on both Macintoshes and IBM PCs, this reduces confusion by keeping the interface uniform across both platforms. Although neither the hardware nor the standard operating system of PCs supports menus directly, contemporary machines are powerful enough to handle such graphic elements directly from the software.

Another structure familiar to most computer users today is the spreadsheet. Eye Lines uses spreadsheets to enter, store, and edit specifications for both stimulus images and experiments. In EL Draw, the image spreadsheet contains the x , y coordinates of line endpoints, as well as the length and angle of each line. It is a true spreadsheet because it will calculate missing values and recalculate all of an object's coordinates when a change is made to any element of that object.

Representing an image as a spreadsheet of coordinates provides necessary precision, but additional invention was needed to help users relate the numbers to the image they represent. An early version that required the user to switch back and forth between coordinates and image was difficult to use, especially for beginners. In the current version (see Figure 1), the spreadsheet and the image share the same screen, and the user can turn on a feature that coordinates the two views by blinking the image line segment that corresponds to the spreadsheet row in which the cursor is located.

Multilevel flexibility. An important way of reducing the user's cognitive load is to provide as much flexibility as possible regarding how the software is used. It is most important to minimize "tunnels"—that is, rigid sequences that once begun must be followed to completion—where a mistake can require one to repeat the entire sequence. Wherever possible, the user should be allowed to complete tasks in whichever order he or she prefers. Both pull-down menus and spreadsheets help provide this flexibility on a local level, but it is also important to maintain

Figure 1. EL Draw displays image and coordinates simultaneously. Blinkseg command establishes visual link between spreadsheet cursor and blinking image segment. For each segment, the spreadsheet displays: startX, startY, length, angle, endX, and endY. An S or E in the first column links the segment to the start or end of another.

Figure 2. To test image manipulation without leaving EL Draw, the user adds Ls (length) or As (angle) to those segments that are to be adjusted, then chooses the Preview command from the View menu.

flexibility across major functions. For example, in Version 1.0 of Eye Lines, users could move quickly among modules controlling the four functions of drawing images, setting up experiments, collecting data, and sorting data. However, they complained that the system required excessive back-and-forth switching. After an image was created, the user switched to the setup module to specify how it was to be manipulated and then switched to EL Run to see if the manipulation was working properly. If it did not behave as expected, the user had to switch back to either EL Draw or EL Set, or both, sometimes going through several cycles of editing and testing. The problem was the rigid requirement that an experiment be set up before image manipulation could be tested. With Version 1.1 of Eye Lines, the user can test the image manipulation while drawing the image, without having to set up a formal experiment (see Figure 2). More to the point, he or she has the choice of either testing the manipulation while constructing each image or waiting to test the entire experiment as a whole.

A new graphic interface for data configuration. Many researchers have found themselves copying the data output from one program and typing it by hand into a statistics program because the configurations did not match and the reconfiguration utilities were too limited or confusing (Butler, 1988). Eye Lines stores raw data in temporal order. Depending on the design, it might be useful to sort it across as many as four dimensions: trial, subject, cycle (if the experiment cycles the subject through the list of trials more than once), and run (if subjects run the experiment more than once). It is not difficult to devise an algorithm that loads data into a four-dimensional (4-D) matrix. But most data analysis software requires data to be input as a 2-D array. In some cases, this is achieved by collapsing across one or more dimensions to produce a 2-D array (e.g., subjects $\times$ trials). The user must specify which dimensions to collapse, the row dimension, and the column dimension. In other cases, there is a need to represent more than two data dimensions in a single spreadsheet. The user must specify, for both rows

Figure 3. EL Sort displays a movable rectangular icon for each active dimension. In this example, each subject will be represented by a single row. Trials and runs will be arrayed across columns in hierarchical order (T1R1, T1R2, T2R1, T2R2, etc.). The data will be averaged (collapsed) across cycles.

and columns, the dimensions to be displayed and their hierarchical order.

The challenge was to design an interface that gives the user all of the possible choices in a way that is relatively easy to understand and quick to operate. The solution chosen for EL Sort is to present the four possible dimensions as icons that can be moved around a map to represent all possible 2-D output configurations (see Figure 3). When the module is activated, it surveys both the experiment setup file and the raw data file and presents icons for only those data dimensions that have more than one value. To collapse data across a dimension, the user clicks the mouse in a small box to unhighlight that icon. After using the mouse to arrange the icons as desired, a click on the OK button at the top of the screen causes the module to sort the data as indicated. The result is written to an output file that can be read by most statistical and spreadsheet programs. The rearrangeable map allows the users to choose from among numerous possibilities without requiring them to learn numerous codes or arcane syntax.

Context sensitive defaults. One of the most widespread strategies for making complex software appear simpler to beginners is to provide default values for most choices. Where possible, default values should be influenced by context. For example, EL Sort reads the raw data file and presents its icons in different patterns depending on which dimensions are relevant. Another example is that when the user switches from one program module to another, the newly called module uses several criteria to guess which file the user wants to open.

Advanced features hidden. Another strategy is to hide some of the more esoteric options in order to present a less intimidating interface. In EL Set, there are advanced display controls and an expanded spreadsheet for setting up multiple stimulus fields for any trial. As shown in Figure 4, these are hidden on start-up but can be turned on by a single menu selection. As the complexity of a program increases, it becomes progressively more difficult

to keep the interface easy and fun. Some of Eye Lines' advanced features require learning specific codes or syntax. The efforts to maintain simplicity have been tilted toward the basic functions encountered by beginners.

Teaching Experimental Thinking

The central problem in the teaching of experimental psychology has always been how to engage students' minds in the uncertain intellectual challenges of research (Eckerman, 1991). Called "inquiry training" by Posner & Osgood (1980), this is much more difficult than transmitting facts, theories, or techniques. Traditional laboratory exercises, which emphasize technique, seldom allow students to experience the thrill that comes through the process of pondering a mystery, forming a question, and then designing an experiment to answer it. Experiment generators can provide more of this experience in less time by shortening the development cycle.

We have used Eye Lines both in a laboratory course in sensation and perception and as part of a summer student research institute. After a short introduction to the software, students were divided into groups of 2 or 3 and given one of several very simple experiments that measure the strength of the Müller-Lyer, Ponzo, or Zöllner illusions (see Figure 5). After running these on each other and perusing the data, they were asked to begin exploring the illusion by altering the stimulus configuration and testing for changes in illusion strength. After a second

Figure 4. EL Set presents a simple interface until the user needs advanced controls or more than one stimulus per trial.

Figure 5. Screens from EL Run showing a Zöllner experiment designed to be easily modifiable by beginning students.

cycle in which they were free to refine their first experiment or to change direction, they were sent off to read the literature and design a more ambitious project. Although students needed the usual amount of help to interpret journal articles, design their experiments, and so forth, there were almost no cases of students getting stuck or being unable to generate a research idea. The summer research groups produced interesting results that have led to further work (Ferriby, Black, Buschman, & Beagley, 1992; Karvonen, Richardson, Mass, & Beagley, 1992).

Fifteen of the 20 students in last winter's perception class responded to an invitation to comment on the software and its use in the course. Of these, 8 made general comments on their enjoyment of the experience, describing it as "fun," "cool," "interesting," or "amazing." Eight students also commented specifically on the educational value of designing their own experiments or using their own creativity. Although 9 students described Eye Lines as "simple" or "easy" and nobody described

the programs as difficult, there were 7 requests for more support in the form of more lab assistants or an on-line help facility. Perhaps most encouraging were the 3 students who made substantive suggestions for additional program capabilities.

Conclusion

Eye Lines provides both power and ease of use for a previously unavailable paradigm. It follows the tachistoscope packages in attempting to bridge the gap between introductory teaching software and custom-built, no-frills research programs. We need a set of tools, spanning a broad range of paradigms, that will do for psychologists what word processors do for writers: help them to grow from beginners into professionals.

REFERENCES

- ABBOUD, H. A. (1991). *SuperLab* [Computer program]. Wheaton, MD: Cedrus Corporation.
- BEAGLEY, W. K. (1991). Eye Lines: A tool for studying visual perception of size and angle. *Behavior Research Methods, Instruments, & Computers*, **23**, 183-186.
- BUTLER, D. L. (1988). Selection of software in the instructional laboratory. *Behavior Research Methods, Instruments, & Computers*, **20**, 175-177.
- COLBOURN, C. (1992). [Review of Eye Lines]. *Psychology Software News*, **3**, 9-11.
- EAMON, D. B. (1982). CEDATS: A cognitive experimental design and testing system. *Behavior Research Methods & Instrumentation*, **14**, 142-145.
- ECKERMAN, D. A. (1991). Microcomputers in undergraduate laboratory training in psychology. *Behavior Research Methods, Instruments, & Computers*, **23**, 91-99.
- ENNS, J. T., OCHS, E. P., & RENSINK, R. A. (1990). VSearch: Macintosh software for experiments in visual search. *Behavior Research Methods, Instruments, & Computers*, **22**, 118-122.
- FERRIBY, J. A., BLACK, J. L., BUSCHMAN, P. L., & BEAGLEY, W. K. (1992, April). Effects of eye scanning speed on visual illusion decrement. Paper presented at the meeting of the Michigan Academy of Sciences, Arts, and Letters, Mount Pleasant, MI.
- KARVONEN, M., RICHARDSON, E. W., MASS, P., & BEAGLEY, W. K. (1992, April). Decrement transfer symmetry in the Müller-Lyer illusion. Paper presented at the meeting of the Michigan Academy of Sciences, Arts, and Letters, Mount Pleasant, MI.
- LESGOLD, A. (1991). Research methodology in the postinformatic age. *Behavior Research Methods, Instruments, & Computers*, **23**, 109-111.
- MEIKE, B., BHARUCHA, J., & BAIRD, J. (1987). *MINDLAB* [Computer program]. Hanover, NH: Courseware Development Group, Dartmouth College.
- POSNER, M. I., & OSGOOD, G. W. (1980). Computers in the training of inquiry. *Behavior Research Methods, Instruments, & Computers*, **12**, 87-95.
- SCHNEIDER, W. (1988). Micro Experimental Laboratory: An integrated system for IBM PC compatibles. *Behavior Research Methods, Instruments, & Computers*, **20**, 206-217.