Behavior Research Methods & Instrumentation
1978, Vol. 10 (2), 148-153

SESSION IV GENERATION AND CONTROL OF VISUAL DISPLAYS: A SYMPOSIUM

ALAN LESGOLD, *University of Pittsburgh, Presider*

A visual display system for reading and visual perception research

GLEN A. TAYLOR, DAVID KLITZKE, and DOMINIC W. MASSARO
University of Wisconsin—Madison, Madison, Wisconsin 53706

The selection of a computer visual display system suitable for word recognition and reading research is described. The software character generation routines permit flexible definition of character sets. The display software permits control of size scaling and point density of characters being displayed as well as control over the temporal microstructure of presenting and refreshing the displayed text.

The selection of a visual display system for a computerized perception laboratory presents a large number of conflicting choices. There is simply no universally superior system. The selection process must be guided by the establishment of a set of minimum specifications followed by a series of compromises between desired characteristics and the properties of actual hardware. Many of the important factors in the selection of a display system have been discussed in the instrumentation literature of the past decade (Buckley & Gillman, 1973; Mayzner, 1968; Scanlan, 1975; Sperling, 1971a, 1971b; Van Gelder, 1972). Before briefly reviewing these factors, it is necessary to enumerate the primary display types available. First are the nonelectronic display systems, such as slide projectors and conventional look-in and projection tachistoscopes. The largest class of devices uses the cathode-ray oscilloscope (CRO). CRO devices are distinguished by whether the cathode-ray tube is a storage or nonstorage tube, whether the beam is independently positioned during display operations or swept in a fixed pattern (raster), and by the extent to which the hardware is capable of performing elemental display operations independently. Thus, CRO display devices range from the simplest display monitors to expensive graphics terminals with considerable local storage. Graphic displays are controlled by microprocessors and possess specialized hardware to support their repertoire of graphics capabilities. Another class of devices is non-CRO electronic displays. Two such devices are the plasma display panel (Scanlan, 1975) and light-emitting diodes (LED) (Nealis, Engelke, & Massaro, 1973).

Funds for the acquisition and development of this system were provided by the National Institute of Education, Department of Health, Education, and Welfare, through its support of the Wisconsin Research and Development Center for Cognitive Learning. The authors express their appreciation to Jack Wende, Rod McCoy, and Eric Pierson, whose contributions to the acquisition, installation, and continued support of the computer hardware cannot go unacknowledged. Requests for reprints should be addressed to Glen A. Taylor, Wisconsin Research and Development Center for Cognitive Learning, 1025 W. Johnson Street, Madison, Wisconsin 53706.

Factors to be considered in evaluating display systems may be grouped into four categories: (1) complexity of the visual display, (2) display frame rate, (3) control over physical stimulus properties, and (4) cost. To the extent that it is necessary to present complex visual stimuli, such as finely detailed pictures or large amounts of text, the choice of a display system becomes constrained. Plasma panels, LEDs, and CRO devices driven by digital-to-analog (D/A) converters compose their displays from primitive elements consisting of discrete points and/or line segments. These elements usually have a uniform "thickness" and often a minimum interelement separation. Typically, the greater the number of these elements necessary to produce a single display frame, the greater the burden on system resources such as memory space and computation time. By contrast, a simple slide projector is capable of presenting a scene of any complexity up to the limits of photographic resolution with no variation in effort.

Display frame rate consists of two related but slightly different components. The first is the minimum amount of time needed to display a single frame. Look-in tachistoscopes and LED displays are parallel devices.

All portions of a single display frame are simultaneously available as soon as the tachistoscope lamps' or LEDs' rise time is completed. CRO and plasma panel displays are sequential devices; they simulate parallel presentation by displaying a large number of elements in a small amount of time. However, there are limits to the number of elements presented per unit time. Displaying a single frame often takes a nonnegligible amount of time. For these devices, the minimum time needed to display a single frame is a function of the speed of the device and the complexity of the frame. Also, for nonstorage CROs the entire display is never simultaneously visible.

The second component of frame rate is the time needed to switch between frames. A slide projector's tachistoscopic shutter might open in 1 msec, but the device typically is unable to present a new frame for hundreds or thousands of milliseconds. CRO displays may take milliseconds to display a single frame, but usually are capable of beginning a new frame within microseconds. Thus, maximum display rate is a function of both the minimum display time and the minimum interframe time. For some applications, the effective display rate is also affected by the total number of frames that must be presented contiguously. Conventional display devices such as tachistoscopes minimize interframe delays by duplicating the display system, but usually only up to a maximum of three or four fields. Electronic display devices offer the potential of displaying N-fields, but there are still limitations on the number of frames that can be stored precomputed in memory. If more frames must be displayed than can be stored, then the frame rate is reduced while frame descriptions are read from external storage, or their computation time is added to display time.

Control over the physical properties of the display is a distinct concern in selecting a display system. Depending on the nature and purpose of the research, it may be necessary to exercise considerable control over such physical characteristics as linear resolution, luminance, figure-ground contrast, color spectral characteristics, and so on. For devices that only simulate parallel presentation, it is necessary to control the presentation of elements within a frame to minimize potential perceptual problems, such as flicker (cf. Sperling, 1971b). A final consideration is the available figure-ground relationships. Electronic display media tend to present light figures on dark ground. Some devices permit presentation of dark figures on light ground, but usually at the expense of some other dimension.

The final concern is cost. The primary consideration is the direct cost of the hardware and the continuing costs of consumable supplies and hardware maintenance. A second consideration is indirect costs for software development and software maintenance. Another less visible cost is the load that the display system places on the laboratory computer in competition with other tasks that must be performed.

Our laboratory is dedicated to studying visual processes in reading and word recognition. Stimuli range in complexity from a single character to perhaps a few hundred characters. Because our research often employs visual masking paradigms, it is necessary to be able to display a frame as quickly as possible, on the order of 1 or 2 msec, and to be able to begin a new frame immediately. For some reading experiments, it is necessary to display frames continually, changing frames every few milliseconds. This presentation rate may need to be maintained for thousands of consecutive frames. Our research demands that displays emulate the appearance of printed text to the greatest extent possible. This means crisp, well defined letters of good contrast. Control over luminance levels is also necessary. Finally, initial cost had to be kept as low as possible.

It should be apparent that no display device mentioned fully satisfies all of these criteria. Although nonelectronic display devices permit the desired control over the physical properties of a single frame, they do not permit the continuous rapid presentation of hundreds of frames necessary for the reading research. Non-CRO devices are costly. LED displays compare unfavorably with CRO devices for flexibility of character generation. Narrowing the selection down to CRO devices, the importance of the display criteria had to be weighed carefully. For instance, storage CRO devices and raster-scanned CRO devices are unsatisfactory because the ability to change display frames rapidly is more important in our work than the ability to present extremely large amounts of text, or text with dark letters on a light background. Storage CROs are unable to change displays rapidly because screen erasure requires a few hundred milliseconds, and raster-scanned CROs present frames (refresh the display) at a fixed and unchangeable rate, typically 60 Hz (16.7 msec). Our final selection is a simple display monitor (Tektronix Model 604) interfaced to the computer with D/A converters. All displays are created as sequences of points presented under direct CPU control. Software for character set generation and display plotting maximizes the flexibility of the display system. The remainder of this paper describes the hardware and software configuration and its advantages and disadvantages.

THE HARDWARE CONFIGURATION

The laboratory computer is based on the DEC LSI-11, which executes the PDP-11 instruction set. Other system components that directly influence visual display capabilities are 20K words of semiconductor memory (16-bit words), two 12-bit D/A converters modified to serve as a "scope driver," a programmable clock that permits interval timing without CPU involvement, and dual high-density floppy-disk drives that provide 1.2 megabytes of on-line storage accessible via a direct memory access channel. The system runs the DEC RT-11 operating system, permitting programming in MACRO-11 assembler language and FORTRAN IV.

The four Tektronix 604 monitors are driven in parallel from the D/A interface. One D/A channel is used for x-coordinate positioning and the other for y-coordinate positioning. This interface has been modified so that loading the y-coordinate D/A channel initiates a one-shot intensification of the z axis of the display monitors. This intensification pulse is delayed 5 microsec after loading a new y-coordinate value to allow for settling of the analog level. The pulse duration is 5 microsec. The maximum point-plotting rate for the hardware configuration is determined by the time required to transfer an (x,y) coordinate pair to the two D/A converters. The move instruction that transfers either an x or y value from memory to a D/A register requires approximately 6 microsec. Thus, a minimum of 12 microsec is needed to display a given point, yielding a maximum point-plotting rate of 83 points/msec when the CPU is dedicated entirely to transferring (x,y) pairs.

THE SOFTWARE

Since the hardware is capable of displaying at most 83 points/msec with total CPU involvement and a pair of memory locations allocated for each point to be displayed, it might seem preferable to have selected a display device which possessed local memory and hardware character generation, thereby saving considerable system resources. However, hardware character generators usually generate characters by illuminating points in a rectangular grid often seven rows by five columns or nine rows by seven columns. Our choice was determined by our desire to have the character set appear as natural as possible and by a need for flexibility. Curved and diagonal strokes should be accurately approximated. Character size and intercharacter spacing should be under the control of the user. Grid systems, especially small grid systems, typically lack these characteristics. A 5 by 7 font employs about 20 points to present the average character. The position of these points relative to each other is determined by the coordinate geometry of the grid rather than the type of line segment they are intended to convey. Our average character (see Figure 1) usually contains nearly twice as many points and the points are evenly spaced around the outline of the letter form. A user can choose to impoverish these characters to have as few or fewer points than a 5 by 7 font and still retain greater legibility because of better point spacing.

The software package consists of three modules depicted diagramatically by the rectangular boxes in Figure 2. The balloons in the figure represent data structures operated upon and produced by these modules. The three modules—the character translator, the display initializer, and the display driver—are functionally independent. The remainder of this section describes the data structures, the modules that transform them, and the flexibilities inherent in this system.

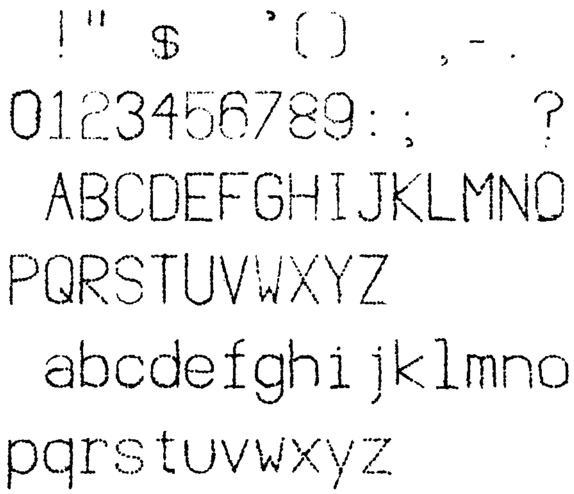

Figure 1. The character set currently being used in our laboratory. The average character is composed of approximately 35 points.

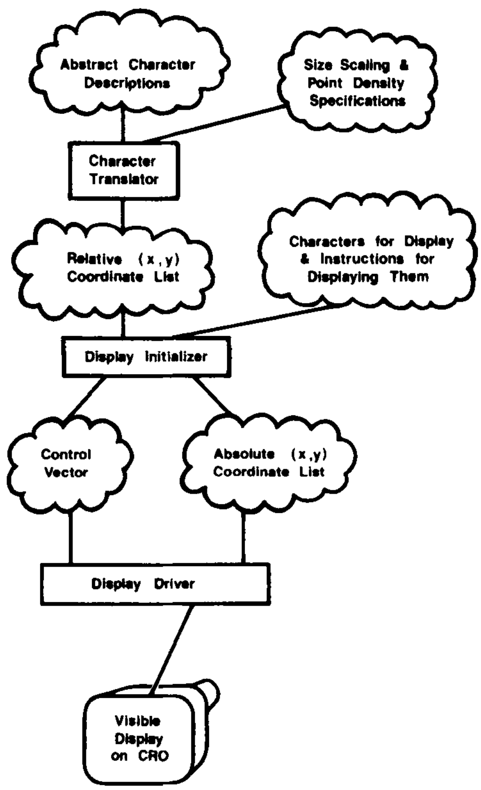

Figure 2. Flow diagram of the display package. Balloons represent data structures used and produced by the three routines that are depicted by rectangular boxes.

Definition of a Character Set

Characters are defined by a user in terms of the strokes required to draw the character. Characters are constructed from straight-line segments, circular arcs of various radii, and points. The strokes defining a character are assumed to be enclosed within an imaginary rectangle whose largest dimension is arbitrarily assigned a distance of 1 unit and which serves as the frame of reference for stroke placement. Figure 3 illustrates this for lowercase a. The figure shows the rectangular enclosure in which our characters have been defined and the seven strokes comprising the letter a. Placement of each stroke is relative to the lower left-hand corner of the rectangle and the length of each stroke is expressed in terms of the "unit" length. The encoded abstract letter form consists of a 1-byte internal name for the character followed by a list of stroke descriptions. For convenience, we have chosen the ASCII encoding for our character set. The description of each stroke consists of a relative (x,y) location for the start of the stroke, the kind of stroke, the stroke's length or arc length, the direction it proceeds from the starting point, and whether or not its endpoints are to be considered part of the stroke or as belonging to a stroke it joins. Although we have only defined the letters, numerals, and punctuation shown in Figure 1, any calligraphy or simple pictographic forms can be defined in this fashion.

The character definitions provide abstract descriptions of the relative size and proportioning of the characters. Two other parameters must be specified before these descriptions can be converted to lists of points to be displayed. The first parameter, the scale value, specifies run-time scaling. This parameter binds the "unit length" to an integer in the range of 1 to $2^{12} - 1$, which is the range of the D/A converters. For example, specifying a scale value of 775 results in a 1-unit line segment appearing as a line 1-in. long on our CRO. Doubling the scale value extends the line to 2 in., and halving it reduces it to .5 in. Since relative proportions are maintained, character size can be manipulated without distortion.

The second parameter specifies how many points are to be spaced evenly along a line of unitary length. This gives the user control over the tradeoff between how closely he wishes to approximate the abstract forms at a given size and the actual number of points comprising each character. The effects of manipulating the two parameters are shown in Figure 4. Considering the a in the center as a standard for comparison, the a on the left has twice the scale value and the same point density. The a on the right has the same scale value but 25% the point density.

It is our usual practice to run the character translator as a stand-alone procedure and retain only the relative (x,y) coordinate list it produces for use during an experiment, much as one compiles a program and retains an object module. There are situations in which it is desirable to include character translation as a step in

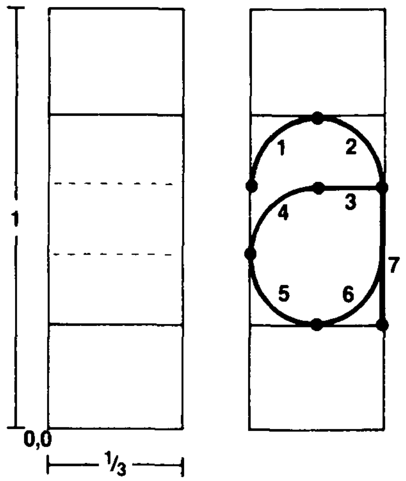

Figure 3. The first panel represents the coordinate space in which our character set has been defined. The second panel illustrates character definition, showing the seven strokes defining lowercase a.

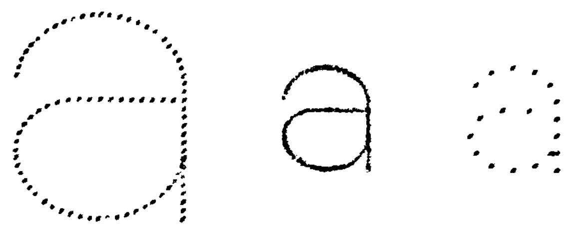

Figure 4. Three representations of the letter a demonstrating the flexibility of size scaling and point density. Taking the middle a as a standard, the character on the left has twice the scale value and the same point density. The character on the right has the same scale value and 25% the point density.

the trial-by-trial display routines, and for this reason the character translator has been written to be as small and as fast as possible. For most character sets, the relative (x,y) coordinate buffer requires considerably more storage than the character translator and abstract character descriptions. If memory space is tight and the extra computation time can be tolerated, trial-by-trial character translation can be effective. A second situation which suggests character translation during run-time display is the need to dynamically modify scaling or point density from trial to trial.

Display Initialization

Display initialization is perhaps the most complex of the three display functions. Nearly all of the computation essential to presenting a display frame is done here. Display initialization establishes (1) where a given display character appears on the CRO screen, (2) the sequence in which the points of the display are plotted, and (3) how long it takes to display the frame.

Transformation of relative (x,y) coordinates to absolute coordinates suitable for plotting requires the addition of an x reference and y reference to each point in a character. Because the relative coordinates for a character are expressed as offsets from a common origin, it is sufficient to bind that origin point to a location on the CRO screen to correctly locate a character. The user supplies the names of the characters to be displayed and specifies their placement in one of two ways. In a simple version of the initialization routine, the characters are assumed to form a single string which the run time horizontally centers on the CRO screen allowing a half-character space between characters. For applications requiring more flexible control over placement, a vector of character origin positions is passed in a COMMON block. Characters to be displayed are then passed as pairs, the character and the numeric index of a character position in COMMON.

Once an absolute (x,y) coordinate buffer has been constructed for the characters, it might appear that initialization is complete. However, it is usually not satisfactory to merely plot the absolute character buffer in simple linear sequence. For most applications, a single plotting of a frame is insufficient. The display driver is called many times to refresh the display. To insure that displays are treated equally from trial to trial, it is convenient for all frames to be plotted in the same amount of time. Equivalent plotting time occurs only for frames that contain the same number of points. This usually does not happen even for displays with the same number of characters, since characters differ in the number of points they contain. This disparity becomes serious if displays with different numbers of characters are presented. Padding or "dead time" is needed to fill out shorter frames.

The concept of dead time is handled systematically by the display initializer. Dead time, which may be thought of as "invisible points," is added to every character in a display to make the character effectively equal to the most dense character in the character set. Thus, little or no dead time is added for g, a relatively dense character, but considerable dead time is added for i. This insures that displays with the same number of characters plot in the same amount of time. For displays with unequal numbers of characters, the user is allowed to specify "invisible letters." These null letters are simply added dead time equal to the densest character.

The final aspect of initialization is determining the sequence in which the points are displayed and the interleaving of dead time. For this purpose the user specifies a "burst factor," which is the fraction of a character to be plotted as contiguous points without intervening dead time. A burst factor of unity signifies that all points of a given character are to be plotted without interruption; factors greater than one imply that each character is to be divided into that many bursts. The total dead time in the display, including intracharacter padding and invisible letters, is also divided into bursts. By default the initialization routine randomizes the presentation of these bursts and includes each burst once in the display frame. Alternatively, the user may elect to specify the presentation order. This permits any sequence of bursts to be obtained, including the repetition of some bursts within a frame. This option makes it possible to refresh some characters or portions of characters more often than others. Since observed brightness is directly related to refreshment rate for frames of short duration, this allows variations of brightness within a display.

Because the temporal resolution of the eye is poor for events of very brief duration, it is possible to paint individual points and give the appearance of coherent text. It is also possible to vary brightness by refreshing the display more frequently (Bloch's law). Without hardware control over intensity, which we presently lack, refreshment rate is the only available real-time control over brightness.

The Display Driver

The display driver is the simplest of the three display modules. The maximum point-plotting rate is only obtained when nothing but load instructions for the two D/A channels are being executed in alternation. A loop of some form is necessary so as not to have a display routine as long as the longest display buffer, but this creates overhead time. This overhead is minimized by removing all conditional tests from the display driver. The display initializer creates a control vector in setting up a frame. This control vector contains a control element for each burst to be displayed in a frame. A control element consists of two parts. The first is either a pointer to the display buffer for a visible burst or a loop count for a dead-time burst (dead time consisting of a tight loop without loading D/A channels; see Figure 5). The second point is an offset into the list of display instructions or to the no-operation (NOP) loop.

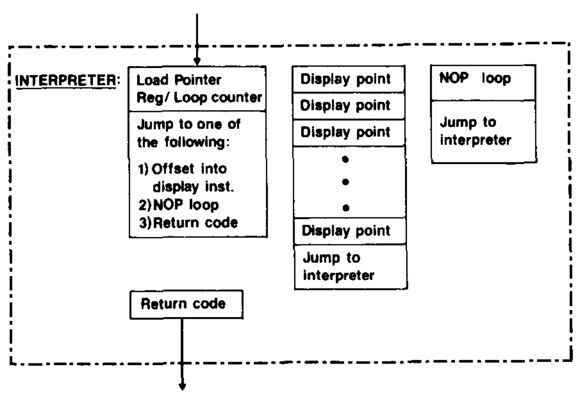

Figure 5. The internal structure of the display driver is represented. Control vector elements are decoded by the "interpreter" into jumps to the list of display instructions or to the NOP loop, as appropriate, to control the presentation of a frame.

The skeleton of the display driver is shown in Figure 5. When the routine is called, a small section of code called the "interpreter" is entered. The interpreter decodes the first element in the control vector by loading the register that serves either as a display buffer pointer or loop counter and blindly transferring control to the place specified in the control element. For a visible burst, this is an offset into the list of display instructions sufficient to plot the appropriate number of points. Otherwise, control passes to the NOP loop. At the end of the burst (visible or invisible), control is returned to the interpreter. Each element in the control vector is decoded, with the final element causing a branch to the return code where a normal subroutine return is made. This externally supplied control structure has several advantages. The display driver is maximally efficient because it makes no decisions. All responsibility for insuring a correct display is placed on the display initializer, which makes the necessary computations before the time critical plotting is begun. Overhead in the display driver is uniform with respect to visible and invisibile bursts, depending only on the total number of bursts.

To summarize, the visual display software consists of three main routines: (1) One converts abstract character descriptions to relative (x,y) coordinate points, (2) another takes a string of characters and instructions about their placement on the face of the CRO and generates a buffer of displayable (x,y) coordinates and a control vector to control their presentation, and (3) a third interprets the control vector displaying a single frame. The system provides the user with control over the definition of his character set, the adequacy of its representation as actual points on the CRO screen, the temporal and spatial ordering of letter presentations within a frame, and the overall refreshment rate. The price exacted for this control is total dedication of the CPU during plotting cycles, and the allocation of a considerable amount of memory for the necessary (x,y) coordinate buffers and control structures. In terms of the decision criteria outlined at the outset of this paper, this display system maximizes control over certain physical and temporal properties of text displays and does so at very modest initial hardware expense. The system is, however, obviously limited with respect to how complex a display it can present in a very brief time. At present, luminance can only be controlled through refreshment rate. The system places high demands on CPU and memory resources, although we have yet to experience undue interference with the nondisplay needs of our research designs.

APPLICATION OF DISPLAY SYSTEM

It is relatively easy to envision experimental paradigms that require three or four successive displays of a dozen or fewer letters. We have developed a paradigm for reading experiments that utlizes the N-channel display capability. The task involves the continual presentation of frames with minimal interframe intervals for thousands of frames. In one version, the stimulus text, maintained as a file on disk, consists of a prose passage containing approximately 10,000 characters. The visual display may be envisioned as a window the height of a line of text and some number of character positions wide, where this number is a parameter of the display. The text streams past this window moving from right to left in a fashion resembling an advertising marquee. The streaming is, of course, accomplished by displaying a large number of discrete frames/sec shifting the text to the left with each successive frame. The number of frames/sec, their duration, and the amount of lateral displacement on each shift are all specified by the experimenter. For this application, the plotting routine is modified slightly to add the absolute x displacement to each relative x value as it is painted. This delay in binding the x coordinate to an absolute value slows the plotting slightly, but the lost plotting time is more than compensated by a reduction in initialization time and a conservation of memory space. The characters in the display are plotted directly from the master list of relative (x,y) pairs for the entire character set without constructing an intermediate absolute display buffer. Of course, there are still limitations on how much text can be displayed in a given period of time. As the window size increases, minimum frame duration also increases. Characters about 35 points/letter can be set up and displayed in approximately 1 msec/character, independently of the number of letter positions that are shifted each frame. At this rate a window of 25 letters may be presented every 25 msec with barely noticeable flicker. With this technique, a truly N-channel tachistoscope is achieved.

REFERENCES

Buckley, P. B., & Gillman, C. B. The slide projector as a computer-operated visual display. *Behavior Research Methods & Instrumentation*, 1973, **5**, 104-106.

Mayzner, M. S. The research potential of a computer-based cathode-ray tube display system. *Behavior Research Methods & Instrumentation*, 1968, **1**, 41-43.

Nealis, P. M., Engelke, R. M., & Massaro, D. W. A description and evaluation of light-emitting diode displays for generation of visual stimuli. *Behavior Research Methods & Instrumentation*, 1973, **5**, 37-40.

Scanlan, L. A. A general-purpose, computer-generated, multi-field tachistoscope. *Behavior Research Methods & Instrumentation*, 1975, **7**, 113-114.

Sperling, G. The description and luminous calibration of cathode ray oscilloscope visual displays. *Behavior Research Methods & Instrumentation*, 1971, **3**, 148-151. (a)

Sperling, G. Flicker in computer-generated visual displays: Selecting a CRO phosphor and other problems. *Behavior Research Methods & Instrumentation*, 1971, **3**, 151-153. (b)

Van Gelder, P. CRT displays in the experimental psychology laboratory. *Behavior Research Methods & Instrumentation*, 1972, **4**, 102-103.