DOT/FAA/AM-92/12

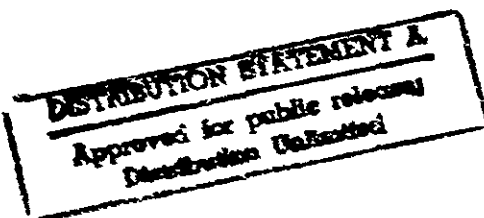Office of Aviation Medicine
Washington, D.C. 20591

A New Test of Scanning and Monitoring Ability: Methods and Initial Results

AD-A249 123

A.M. Revzin

P.G. Rasmussen

Civil Aeromedical Institute
Federal Aviation Administration
Oklahoma City, Oklahoma 73125

March 1992

DTIC
ELECTE
APR 2 1 1992
B

Interim Report

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

U.S. Department
of Transportation
**Federal Aviation
Administration**

92 4 20 094

92--10089

NOTICE

This document is disseminated under the sponsorship of
the U.S. Department of Transportation in the interest of
information exchange. The United States Government
assumes no liability for the contents or use thereof.

1. Report No. DOT/FAA/AM-92/12	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A New Test of Scanning and Monitoring Ability: Methods and Initial Results		5. Report Date March 1992
		6. Performing Organization Code
		8. Performing Organization Report No.
7. Author(s) A. M. Revzin, Ph.D. and P.G. Rasmussen, M.B.A.		
9. Performing Organization Name and Address FAA Civil Aeromedical Institute P.O. Box 25082 Oklahoma City, OK 73125		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Office of Aviation Medicine Federal Aviation Administration 800 Independence Avenue, S.W. Washington, D.C. 20591		13. Type of Report and Period Covered
		14. Sponsoring Agency Code

15. Supplementary Notes
This work was performed under Task AM-A-91-PHY-146.

16. Abstract

Most tasks in the FAA's Air Traffic Control (ATC) system involve long duration scanning and monitoring for continuously changing events occurring within a large visual space. Errors occur, so it is important to understand the causes of such errors to minimize or eliminate them by *changing task design or improving personnel selection. This study* describes a new system for testing scanning and monitoring abilities. The system, as currently implemented, is basically a character identification task. The characters are presented at random intervals and locations within two or more "WorkAreas." The WorkAreas are defined as rectangular areas on a microcomputer display screen. They are filled with a constantly changing random dot pattern and may be located anywhere on the screen.

The subject's task is to press a designated key on the computer keypad when a specified target character appears. Parametric manipulations can evaluate the effects on performance of many variables, including angular separation of WorkAreas, differential workloads in the WorkAreas, and effects of visual noise.

We found a highly significant performance decrement as a function of increasing angular separation of WorkAreas. This is congruent with prior studies, which we interpret as a validation of our test procedure. We did not find practice effects, fatigue effects, or selective attention effects between WorkAreas.

17. Key Words Vision Requirements Scanning/Monitoring Medical Certification Air Traffic Control	18. Distribution Statement Document is available to the public through the National Technical Information *Service, Springfield, VA 22161*

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 20	22. Price

Form **DOT F 1700.7** (8-72)   Reproduction of completed page authorized

ABSTRACT

Most tasks in the FAA's Air Traffic Control (ATC) system involve long duration scanning and monitoring for continuously changing events occurring within a large visual space. Errors occur, so it is important to understand the causes of such errors to minimize or eliminate them by changing task design or improving personnel selection. This study describes a new system for testing scanning and monitoring abilities. The system, as currently implemented, is basically a character identification task. The characters are presented at random intervals and locations within two or more "WorkAreas". The WorkAreas are defined as rectangular areas on a microcomputer display screen. They are filled with a constantly changing random dot pattern and may be located anywhere on the screen.

The subject's task is to press a designated key on the computer keypad when a specified target character appears. Parametric manipulations can evaluate the effects on performance of many variables, including angular separation of WorkAreas, differential workloads in the WorkAreas, and effects of visual noise.

The results of the initial experiments are presented. We found a highly significant performance decrement as a function of increasing angular separation of WorkAreas. This is congruent with prior studies, which we interpret as a validation of our test procedure. We did not find practice effects, fatigue effects or selective attention effects between WorkAreas. The significance of these results is discussed.

INTRODUCTION

This research was undertaken to examine some factors that could contribute to scanning and/or monitoring errors in the ATC system. Most tasks in the ATC system involve scanning for and monitoring discrete events, or targets, occurring over a wide visual angle. Individual air traffic control specialists (ATCSs) may sit in front of a visually complex sector suite subtending an overall visual angle of 120 degrees or more, depending on the position of the ATCS's head relative to the screen. Scanning and monitoring over so wide an area is difficult, and errors do occur. However, though scanning or monitoring problems were often mentioned in conversations we have had with ATCSs, such problems are seldom discussed in standard aviation references (1), though they have received considerable attention in the process control literature (2).

Personal observations and conversations with instructors at the Federal Aviation Administration Academy suggest that both errors of detection and errors of recognition occur. A typical detection error occurs when an ATC fixes attention on, or "locks" onto, one very small display area and does not respond to anything else on the display. A recognition error occurs when an ATCS does perceive a target, but responds improperly. Such errors are usually caused by the complex behavioral factors controlling attention and set (e.g., boredom, fatigue, workshift changes, illness, drugs, medications, environmental toxicants) (3). However, factors related to the spatio-temporal distribution of information on the display may synergically interact with these. For example, the probability of a detection error may be influenced by the number, spacing and relative velocities of targets or by the required number of operational decisions per unit time.

Since scanning/monitoring errors are perceived as a problem by ATCSs, we initiated an investigation of the factors contributing to such performance errors. The first group of experiments examined the effects of varying the spatio-temporal distribution of information on the display. We began with the working hypotheses that scanning or monitoring errors would increase as: (a) target separation increased; (b) the number of targets or target areas increased; (c) the rate of change of target area components

1

increased; (d) the amount of visual noise, or distraction, present increased. (Note: in much of what follows we have grouped these factors under the term "workload," for brevity.) These hypotheses were based on the truism that any scanning task requires transfers of visual attention among the targets and target areas monitored. Each transfer takes time, up to 1 second, depending on task structure (4, 5). As workload increases, the percentage of the ATCS's time spent transferring attention among targets also increases. Thus, relatively less time is available for stimulus detection and recognition and errors will tend to increase.

This paper describes the methods developed and reports initial results on the effects of WorkArea separation on scanning error. We have also appended a full listing of the program implementing our test structure because we feel that this approach has many advantages in evaluating vision related perceptual problems in aviation.

METHODS

General Approach

The purpose of scanning or monitoring is usually the detection of and response to some defined symbol, event or relationship, termed here a "target." A WorkArea, as used here, is a microcomputer-driven display sub-area in which some defined target or set of targets is embedded. As in other long term (>60 minute) studies of scanning and monitoring (2), the variables to be examined include the number of targets per WorkArea, the number of WorkAreas to be searched, the WorkArea size, the separation between or among WorkAreas, the task complexity within each WorkArea and the relative task complexity between or among the WorkAreas.

Selection of the stimulus configuration defining a workarea for studies of scanning was

difficult. For example, any study of the factors affecting an ATCS's ability to scan and monitor must include some kind of character search paradigm since ATCSs must be able to read alphanumeric information displayed at many locations on their sector suite or display. Character search activity can be modeled in several ways. The simplest and most direct is to look for one character (e.g., "A") in an array of characters. The array is usually in a rectangular format and looks something like this:

```
QQRGXXTB
BROOOOOX
DFOOAOOK
QWOOOOOH
ZXBRPBBW
```

The "A" tends to pop out of the field as set up here, but the ability to detect the "A," or any other character, actually depends on the character's position in the array and on the shape of the surrounding characters (6). However, the target should be presented free from any consistently located distracting shapes or patterns that can affect its readability since such readability changes could confound changes due to the desired variables. It is also desirable to study monitoring efficacy in the presence of visual noise, since noise is always present on an ATCS display. The challenge was to devise a character identification task with noise and without the kind of pattern interference problem mentioned above.

The method developed uses two or more constantly changing random dot WorkAreas into which the characters are written; the random dot display giving constantly changing background noise without patterns to interfere with the character identification. Producing rapidly changing random dot patterns presents problems. The investigators who first used the technique employed special hardware built into a minicomputer (7). Others have used very fast minicomputers or "workstations" with fast graphics capabilities (B. Frost, York University, Canada, personal communication). These are extremely

2

effective approaches but beyond the resources presently committed to this project. Therefore, we developed a relatively simple software method of generating random dot arrays using an "8 MHz" IBM PC-AT™ with an EGA adapter and color display. A listing of the Borland Turbo Pascal 5.5™ program used for these experiments is appended. The resulting software controlled random dot display runs slowly, limited by the compiler, processor and graphics board used. The slow running speed, in turn, limited both the WorkArea size and the number of Work-Areas that could be used. However, this approach can give useful results, used within its limits.

For the first studies, we used two Work-Areas; rectangular screen areas filled with a constantly changing random dot array. The random dot writing process continually over-wrote (i.e., erased) any characters displayed in the WorkArea. Thus, a character's legibility was maximum when first displayed and deteriorated over time. This use of the random dot writing process to slowly erase the displayed symbol is unusual and, perhaps unique, as we have not found similar approaches in the litera-ture. When targets are presented in this way, the subject (S) must respond relatively quickly to a target presentation lest the symbol become illeg-ible or changed enough to be confused with other symbols (e.g., as written on the screen, "8" and "3 can be confused after replacement of one or two pixels).

Workload can be easily varied by chan-ging parameters such as the dot density within the WorkArea, the rate of presentation of the characters, the size of the set of displayed char-acters and the probability of occurrence of the defined "target character."

Figure 1 is a reverse contrast full-scale printout of two rectangular WorkAreas separ-ated by 7 millimeters (mm). It illustrates the main test display characteristics noted above. However, contrast was reversed (black on white rather than white on black) to improve print-out legibility. Note the "3" near the center of the left WorkArea. It was written 0.5 seconds be-fore the frame was "frozen" for printout. It is still clearly legible. The top center of the left WorkArea has the remains of another "3," written 2.0 seconds before the first "3." The remains of a "5," written at the same time, can be seen in the bottom right corner of the right WorkArea.

The pixel writing process replaces about 720 or 7% of the pixels in each WorkArea each second, limited mostly by the speed of the EGA adapter. Our preliminary measurements suggest that a symbol remains clearly recognizable for 1.5 ± 0.5 seconds, with some fragments visible for 6 ± 2 seconds. The times vary due to the vagaries of the random number driven pixel placement process. Thus, as shown in Figure 1, the remains of one or more symbols may be present when a new one is displayed. Since the minimum interval between character presen-tations is 2.25 seconds, there is always a clear difference between the newly displayed symbols and the "older" ones. These old symbols do, however, constitute another visual "noise" factor on the display. Since these noise locations vary randomly over time, no consistent pattern is present and the readability problems mentioned earlier do not seriously obtrude.

Implementation

Two WorkAreas are present. Each is a 100 by 100 pixel array, or 10,000 pixels per WorkArea. The WorkArea's on-screen size is 31 mm by 31 mm. At the normal viewing dis-tance of approximately 60 centimeters (cm), each WorkArea subtends 3.3 degrees of arc and each pixel subtends 2 minutes of arc. In fact, the angular sizes are approximate because a subject's eye to screen working distance does vary somewhat through any test session. Vari-ous systems to control head movement were considered, and tested, but discarded as too cumbersome or uncomfortable. Rather, head
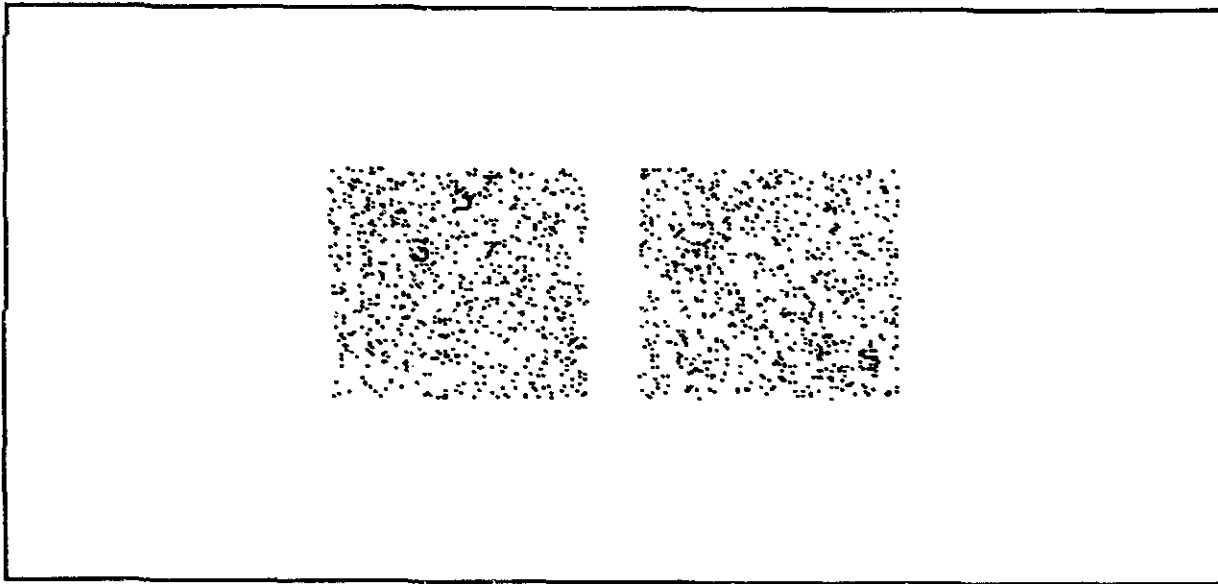
3
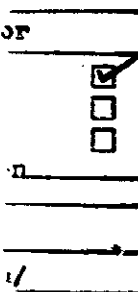
Figure 1

movements were minimized by suitable location of S's seat, the keyboard, the display and a mask surrounding the display. Direct observation of Ss indicate that this was fairly successful, with actual eye to screen distances ranging from 40 cm to 70 cm. The individual test sessions were designed to run for 80 minutes with an additional 10 minutes allotted for administrative details.

The character set used was "23580." These numerals were selected after preliminary evaluation and extensive literature review (6). The "5" was selected as the target symbol to simplify the program but only after preliminary experiments indicated that, in this system, the readability of all of these numbers was equal. The target symbol was constantly displayed to the Ss in a cue line (i.e., "The target symbol is 5.") at the top center of the display (not shown in Figure 1). All symbols were presented at randomly selected locations within the WorkArea and at some average interval, randomly varied. The probability of occurrence of the target symbol was separately programmable. The symbol presentation interval in each WorkArea averaged 3 seconds with a programmed random variation of plus or minus 0.75 seconds. This

interval and the pixel replacement rate combine to give a moderate workload: the Ss could do the test adequately, but constant attention was required and it was not possible to work without errors.

The task of the Ss was to monitor the WorkAreas for the appearance of the target numeral. If the target appeared in the left WorkArea, S pressed the left arrow key on the keypad. If the target symbol appeared on the right, S pressed the right arrow key. Thus S's "perception to response" coupling was simple, "right side to right key" and "left side to left key." The intent here was to make the system so simple to learn that training time would be negligible and little practice effect would be seen within each test session. This intent was realized as shown by the data cited below.

As implemented in the program, two types of errors could be made by the Ss. An error of commission, or of recognition, occurred when S depressed a key when there was no target symbol on the key side or if S responded more than once to the same target presentation. An error of omission, or detection, occurred if an S did not press the proper key in the interval

between the appearance of a target symbol and the appearance of the next symbol in that area. The preliminary analyses reported below, however, consider only the percentage of correct responses made. A fuller analysis, considering detection and recognition errors and any "locking" phenomena will be combined with later data and prepared for publication in a subsequent report.

In these experiments a constant and equal workload was used in each of the two WorkAreas. The WorkAreas separations were 3 mm, 42 mm, 84 mm, and 126 mm (0.3, 4.7, 9.3 and 14 degrees of arc, respectively), inner edge to inner edge.

Each subject was tested in one session lasting for 80 minutes. Thirty two Ss were used, 19 to 40 years of age. The Ss were volunteers recruited and paid by a local contractor. None had prior ATC experience. We measured each S's vision using a VisTech ™ visual contrast sensitivity test system. Each S was also tested for "right eye" or "left eye" dominance so that any positional effects due to eye dominance or "handedness" could be measured. The age and sex of each S were also noted. All this required about 10 minutes. The S was then taken to the testing setup and was given a brief but thorough introduction to the test, during which we explained the purposes of the study and the structure of the task.

Each test session was divided into 4 segments of 20 minutes each. There was a 3 second interval between segments and no "breaks" in the session were permitted. The WorkArea separation differed from segment to segment. The order of presentation of the separations was determined by a balanced latin square experimental design. That is, each S worked at all 4 separations and all separations were present in equal numbers. Thus each S's performance could be evaluated in terms of order of presentation and/or WorkArea spacing, and performance could also be evaluated for practice or fatigue effects.

The software program created separate individual data and performance disk files for each S. For analysis, the text files were appended to a database file and parsed to form manipulable data structures which could then be passed to a spreadsheet or a statistical program for detailed analysis.

Data was analyzed using standard statistical procedures provided in the "Number Cruncher Statistical System ™" (NCSS) program. Since the raw data distribution was skewed, non-parametric analyses were normally used, predominantly the Wilcoxon matched pairs two-tail test. More conventional t-tests were also used.

RESULTS and DISCUSSION.

Initial results from the 32 Ss are presented in Table I. The data are presented as the percentage of correct responses for each S at each each separation and within each segment, together with the means, standard deviations, maxima and minima for each category. Note that the operation of the random number generator controlling symbol presentation is such that in any given segment, the number of neutral symbols differs among the Ss. Therefore we used percentages to ensure data comparability among Ss.

In Table I the averages for percentages correct responses at various separations (in the line labeled MEAN) suggest that the mean percentage of correct responses decreased as a function of WorkArea separation. These raw numbers also suggest that, though the best performance was at a separation of 0.3 arc degrees, there was little difference in performance at the larger separations. Further, the mean percentage of correct responses did not seem to differ significantly among the test segments. The data was evaluated for significance using the non-parametric Wilcoxon Test (8) and conventional t-tests. The results were identical.

5

TABLE I

Percent Correct Responses for each Subject
By separation (degrees) and by segment.

	SEPARATION				SEGMENT			
SID*	0.3	4.7	9.3	14.0	1	2	3	4
A119	87.4	87.7	83.8	83.1	87.4	87.7	83.8	83.1
A115	95.4	93.1	91.3	88.6	95.4	93.1	91.3	88.6
A111	92.9	86.5	79.3	77.8	92.9	86.5	79.3	77.8
A131	94.7	93.7	90.8	90.8	94.7	93.7	90.8	90.8
A127	79.1	80.0	77.1	71.3	79.1	80.0	77.1	71.3
A123	92.6	88.5	93.7	83.6	92.6	88.5	93.7	83.6
A107	76.9	74.7	60.1	54.0	76.9	74.7	60.1	54.0
A103	91.1	94.4	88.3	90.1	91.1	94.4	88.3	90.1
A126	87.7	82.7	84.9	88.4	82.7	88.4	87.7	84.9
A114	72.5	71.6	57.0	65.4	71.6	65.4	72.5	57.0
A122	86.3	76.9	77.2	78.4	76.9	78.4	86.3	77.2
A118	85.3	79.4	71.0	78.7	79.4	78.7	85.3	71.0
A102	72.5	78.3	56.9	56.6	78.3	56.6	72.5	56.9
A130	91.7	80.8	81.9	77.9	80.8	77.9	91.7	81.9
A110	84.6	75.7	79.7	82.1	75.7	82.1	84.6	79.7
A106	80.7	69.4	79.4	68.2	69.4	68.2	80.7	79.4
A101	87.9	89.1	82.9	93.2	82.9	87.9	93.2	89.1
A129	88.8	87.9	86.4	83.9	86.4	88.8	83.9	87.9
A125	87.1	89.9	80.9	86.0	80.9	87.1	86.0	89.9
A121	85.2	80.6	66.5	73.9	66.5	85.2	73.9	80.6
A105	92.3	81.1	76.0	75.9	76.0	92.3	75.9	81.1
A113	86.3	75.5	78.8	78.4	78.8	86.3	78.4	75.5
A109	74.2	59.6	73.4	47.2	73.4	74.2	47.2	59.6
A117	62.5	58.5	50.3	54.8	50.3	62.5	54.8	58.5
A120	88.6	86.3	79.7	82.8	82.8	79.7	86.3	88.6
A116	92.0	85.9	89.0	89.0	89.0	89.0	85.9	92.0
A132	78.3	68.4	64.8	66.3	66.3	64.8	68.4	78.3
A108	86.4	87.6	83.6	81.2	81.2	83.6	87.6	86.4
A104	77.4	62.1	67.1	66.5	66.5	67.1	62.1	77.4
A124	89.7	81.6	80.3	80.0	80.0	80.3	81.6	89.7
A112	71.1	47.1	64.0	64.5	64.5	64.0	47.1	71.1
A128	86.5	68.4	77.8	69.9	69.9	77.8	68.4	86.5
MEAN	84.55	78.86	76.69	75.89	78.77	80.15	78.33	78.74
STDEV	7.78	11.03	10.72	11.60	9.88	10.08	12.53	10.94
MAX	95.43	94.41	93.66	93.17	95.43	94.41	93.66	92.00
MIN	62.50	47.13	50.31	47.22	50.31	56.65	47.13	53.99

* SID = Subject Identification Data Block Number.

TABLE II

STATISTICAL TEST RESULTS
WILCOXON MATCHED PAIRS TEST

PAIRED SEPARATIONS~	p <	PAIRED SEGMENTS	p <
0.3 - 4.7	0.001**	1st - 2nd	0.3
0.3 - 9.3	0.00001**	1st - 3rd	0.8
0.3 - 14.0	0.00001**	1st - 4th	0.8
4.7 - 9.3	0.2	2nd - 3rd	0.8
4.7 - 14.0	0.02*	2nd - 4th	0.9
9.3 - 14.0	0.8	3rd - 4th	0.9

~ = separations in arc degrees.
** = Significant differences.
* = Differences probably significant.

Table II shows analyses for paired segments and paired separations. In each instance, the hypothesis tested was that the first element of the pair was larger than the second. Table II clearly shows that the differences in performance between 0.3 arc degrees and the other separations were highly significant. The data also indicates that performance does not change much as separation increases beyond 4.7 arc degrees. That is, our initial hypothesis was true: performance does become worse as separation increases from 0.3 to 4.7 arc degrees, though, surprisingly, not much change is seen beyond that. The small performance decrement seen at the wider separation surprised us, and warrants further study.

We evaluated possible practice effects by comparing performance in the first segment to that in succeeding segments. Table II shows, as Table I suggests, that there were no significant practice effects: that is, there are no significant differences in performance among the segments. The test was designed to be simple to learn and to do, and this intent was realized. The test was also designed to show some fourth segment "fatigue" effects since preliminary test runs, using CAMI personnel as Ss, suggested

that such effects might occur. However, no such effects were seen. It may be significant that the ages of the CAMI personnel tested were rather greater than the ages of the Ss used in the study. Such an age effect may be an important factor in scanning and monitoring performance and, again, warrants further study (9).

The raw data in Table I shows that performance varied quite widely among the Ss. Performance studies done without reinforcement (i.e., without rewards for good performance or punishment for bad) can be difficult to interpret, since there is no way to control or evaluate a subject's motivation and/or effort during the test. Like most experiments of this kind in the literature, no reinforcements were used for this study. Therefore, we instituted tests, now underway, using identical test procedures and parameters, but adding a simple positive reinforcement: Ss are rewarded with extra money if their performance exceeds the average levels reported here. These experiments should indicate whether reinforcement for good performance will be desirable for future experiments and may indicate whether, or to what extent, the present results may be misleading.

The large variations among Ss are significant in other ways. Limited test-retest data suggests that a subject's performance remains stable over time. If so, it seems possible that a scanning test similar to this could be used in screening ATC personnel for selection or retention. Also, some performance, scanning and monitoring studies are done with small numbers of Ss, frequently less than 10 (2). This raises sampling error concerns. If 10 Ss who score well happen to be selected the results will differ somewhat from studies that happen to select Ss with poor performance. Though 32 Ss is not a large number, the very fact of the variability indicates that the performance distribution in our Ss may be typical of the general population of 19 to 40 year olds.

Preliminary analyses, to be reported in a subsequent publication, suggest that no "locking" occurs, even at the largest separations. This is not surprising. Workloads were nothigh, nor were error rates. Thus,some of the factors which could induce selective attention (2) were missing. Furthermore, the workloads in the WorkAreas were identical, as were the points deducted for each error, so there was no reason for any S to concentrate on one WorkArea rather than the other. The effects of both factors, differential workloads and differential reinforcement values, will be separately evaluated in the experiments to follow.

CONCLUSIONS.

The intent of the current paper is to present a new test of scanning/monitoring performance and to report some initial data showing that application of the method produces meaningful data.

Available resources permit testing of 3 or 4 Ss each day depending on task training and practice requirements for special situations. Thus, we can study large numbers of Ss more quickly and efficiently than if we used more traditional methods which require significant training or practice times. In addition, parametric variations can be introduced readily, as required. Thus, to date, the test seems effective, flexible, and powerful. Even these early results clearly show that performance suffers as the angle scanned increased from 0.3 to 4.7 degrees of arc. This argues, not surprisingly, that ATC's sectors should be kept as compact as possible to minimize scanning errors. Furthermore, there is no evidence of any selective attention effects. Thus, separation, by itself, probably does not induce the "locking" effect.

8

REFERENCES.

1. Hopkin VD. Air Traffic Control. In: Wiener EL, Nagel DC. eds. Human factors in aviation. San Diego: Academic Press; 1988:83-110.

2. Moray N, Rotenberg I. Fault management in process control: eye movements and action. Ergonomics 1989; 32:1319-42.

3. Moray N. Monitoring behavior and supervisory control. In: Boff KR, Kaufman L, Thomas JP, eds. Handbook of perception and human performance. Vol II. New York: John Wiley; 1986.

4. Flowers JH, Lohr DJ. How does familiarity affect visual search for letter strings? Perception & Psychophysics 1985; 37:557-67.

5. Morris A, Temme LA. The time required for U. S. Navy fighter pilots to shift gaze and identify near and far targets. Aviat. Space Environ. Med. 1989; 60:1085-9.

6. Boles DB, Clifford JE. An upper- and lowercase alphabetic similarity matrix, with derived generation similarity values. Behav. Research Meth. Instruments & Comput. 1989; 21:579-86.

7. Julesz B. Foundations of cyclopean perception. Chicago: University of Chicago Press; 1971.

8. Bradley JV. Distribution-free statistical tests. Englewood Cliff. Prentice-Hall; 1968.

9. Ball KK, Beard BL, Roenker DL, Miller RL, Griggs DS. Age and visual search: expanding the useful field of view. J. Opt. Soc. Am. A. 1988; 5:2210-19.

```pascal
PROGRAM SCAN15A;

USES
   dos, crt, graph ;
CONST
  Symbol : string[5] = '30532';
  OffSetArray : Array[1..16] OF integer = (280,10,420,140,
                                           140,420,10,280,
                                           10,140,280,420,
                                           420,280,140,10);

VAR
  Sname : string[25];                                        { Name }
  S_Age : string[2];                                          { Age }
  S_ID : string[5];                                          { S's ID }
  S_Sex, Glasses, DominantEye : char;                       { Obvious }
  VCS : string[5];                                         { VCS score }
  year, month, day, dayOfWeek : word;                     { Date info. }
  hour, minute, second, sec100 : word;             { Session start time }
  TimeRef, TimingCtr : Longint;            { PixelLoop Ctr for epoch time }
  SessionCtr : integer;                                    { # epochs }
  F : text;                                              { For output }
  OutFile : string[20];                            { Path & filename }
  GrDriver, GrMode, GrError : integer;        { graphics initialization }
  TargSym : char;                                 { Target character }
  x0, x1, y0, y1 : word;                       { Array output locations }
  k0, k1, k2 : integer;                       { Misc. temp. variables }
  OffSet : integer;                            { Display separation }
  OffSetArrayIdx, OffSetArrayIdx0 : integer;            { OffSet seq. }
  LftX, RtX, LftY, RtY : integer;                   { Array Locations }
  ctr0, ctr1, ctr2 : integer;                        { Misc. counters }
  LftDotProb, RtDotProb : integer;              { "On" pixel density }
  LftRespCtr, RtRespCtr : integer;          { Correct response ctrs. }
  LftErrCtr, RtErrCtr   : integer;                      { Error ctrs. }
  LftSymbolCtr, RtSymbolCtr : integer;            { how many symbols? }
  LftTargSymCtr, RtTargSymCtr : integer;            { TargSym ctrs. }
  RespCtr, TargSymCtr : Longint;             { Ctrs. for "scoring." }
  LftLoopCtr, RtLoopCtr : integer;    { Pixel loop ctrs.-loop freq. control. }
  LftLoops, RtLoops : integer;               { Symbol display freq. }
  SlowDown : integer;                        { Pixel loop slowdown }
  color, nocolor  : word;
  CharSize : word;                                  { Size of chars. }
  sym0, sym1, sym2 : char;                  { Keep track of symbols }
  ch0, ch1, ch2, ch3, chx  : char;          { Utility loc. for chars. }
  LftTargSymOn, RtTargSymOn  :boolean;




{ ----------------------------------- }
PROCEDURE Beeper;
BEGIN
  sound(1000);
  Delay(75);
  nosound;
END;  { Beeper }


{ ----------------------------------- }
FUNCTION D2 (k2:word) : word;                    { For time conversion }
BEGIN
  IF k2 < 10 THEN write('0');
  D2 := k2;
END; { FUNCTION D2 }
```

A-1

```
{ ------------------------------------- }
PROCEDURE StoreConstantData;

BEGIN
{------ Constants for any one session. -------- }
writeln(F, 'NAME:                  ',Sname);
writeln(F, 'AGE:                   ',S_Age);
writeln(F, 'SUBJ. ID:              ',S_ID);
writeln(F, 'SEX:                   ',S_Sex);
writeln(F, 'OPTICS?:               ',Glasses);
writeln(F, 'VCS:                   ',VCS);
writeln(F, 'DOM. EYE:              ',DominantEye);
writeln(F, 'DATE:                  ',D2(day),'/',D2(month),'/',year);
writeln(F, 'START TIME:            ',D2(hour),':',D2(minute));
writeln(F, 'SEGMENT TIME:          ',k1);
writeln(F, 'NO. OF SEGMENTS:       ',SessionCtr);
k0 := OffSetArrayIdx0;
writeln(F, 'OFFSET SEQ.:           ',OffSetArray[k0],' ',
                                     OffSetArray[k0+1],' ',
                                     OffSetArray[k0+2],' ',
                                     OffSetArray[k0+3]);
writeln(F, 'TARGET:                ',TargSym);

writeln(F, 'TARGET SIZE:           ',CharSize);
writeln(F, 'SLOWDOWN UNITS:        ',SlowDown);
writeln(F, 'L. PIXEL DENSITY:      ',LftDotProb);
writeln(F, 'L. LOOP TIME:          ',LftLoops);
writeln(F, 'R. PIXEL DENSITY:      ',RtDotProb);
writeln(F, 'R. LOOP TIME:          ',RtLoops);
writeln;        { Blank line for parsing }
END; { PROCEDURE StoreConstantData }

{ ------------------------------------------------- }
PROCEDURE SetUps;

BEGIN
REPEAT
   clrscr;
   writeln('              < < <    SET CAPS LOCK FOR ALL ENTRIES    > > >');
   writeln;
   writeln;
   write('S. NAME  (25 max.):   ');
   readln(Sname);
   write('AGE   (18..40):   ');
   readln(S_Age);
   write('S. ID   (4 max.):    ');
   readln(S_ID);
   REPEAT
     write('SEX (M/F/X):   ');
     readln(S_Sex);
   UNTIL (S_Sex IN ['m','M','f','F','x','X']);
   REPEAT
     write('OPTICS (N/G/C/X):   ');
     readln(Glasses);
   UNTIL (Glasses IN ['n','N','g','G','c','C','x','X']);
   write('VCS SCORE (5 digits):   ');
   readln(VCS);

   REPEAT
     write('DOM. EYE (L/R/X):   ');
     readln(DominantEye);
   UNTIL (DominantEye IN ['r','R','l','L','x','X']);
```

A-2

```
REPEAT
                                    { - Check to see if file already exists! - }
    write('DIR. & FILENAME (20 max.):  ');
    readln(OutFile);
    Assign(F, OutFile);
    {$I-}
    Reset(F);
    k2 := IOResult;
    {$I+}
    IF k2 = 0 THEN
    BEGIN
    Beeper;
    writeln;
    writeln('<<< FILE ALREADY EXISTS >>> ');
    writeln;
    writeln('OVERWRITE EXISTING FILE?  Y = YES,  N FOR NEW FILE');
    readln(ch0);
    writeln;
    END; { IF }
UNTIL (k2 <> 0) OR (ch0 = 'y') OR (ch0 = 'Y');
 Assign(F, OutFile);                                    { OK. ReAssign File }
 Rewrite(F);                                            { & prepare for writing }
 GetDate(year, month, day, dayOfWeek);
 GetTime(hour, minute, second, sec100);
 REPEAT
 write('SEGMENT TIME (1..20 min.):  ');
 readln(k1);                                 { k1 is a scrap integer location }
 UNTIL (k1 IN [1..20]);
 TimeRef := k1*(60000000 DIV 574);     { Sets Pixel loop TimeCtr reference }
 REPEAT
 write('OFFSET SEQ. (1..4):  ');
 readln(k0);
 UNTIL k0 IN [1..4];
 REPEAT
 write('NO OF SEGMENTS (1..4):  ');
 readln(SessionCtr);
 UNTIL SessionCtr IN [1..4];
 CASE k0 OF
 1 : OffSetArrayIdx := 1;
 2 : OffSetArrayIdx := 5;
 3 : OffSetArrayIdx := 9;
 4 : OffSetArrayIdx := 13;
 END;  { CASE }
 OffSetArrayIdx0 := OffSetArrayIdx;                             { set for data out }
 REPEAT
    write('TARGET SYMBOL (8, 0, 5, 3, or 2):  ');
    readln(TargSym);
 UNTIL (TargSym IN ['8','0','5','3','2']);
 REPEAT
    write('TGT. SIZE (4..6):  ');
    readln(CharSize);
 UNTIL (CharSize IN [4..6]);

 REPEAT
    write('SLOW PIXEL CHANGE RATE? (Y/N):  ');
    readln(ch0);
 UNTIL (ch0 IN ['n','N','y','Y']);
 IF (ch0 = 'y') OR (ch0 = 'Y') THEN
    BEGIN
      REPEAT
        write('SLOWDOWN CONSTANT (1..10):  ');
        readln(SlowDown)
```

```
      UNTIL (SlowDown IN [1..10]);
    END
  ELSE
    SlowDown := 0;
  write('ALL ENTRIES CORRECT?  Y = YES, N = REPEAT SCREEN:   ');
  readln(ch0);
UNTIL (ch0 = 'y') OR (ch0 = 'Y');
REPEAT
  clrscr;
  writeln('ENTER LEFT ARRAY VARIABLES -');
  writeln;
  REPEAT
    write('    L. PIXEL DENSITY (8..40) :    ');
    readln(LftDotProb);
  UNTIL (LftDotProb IN [8..40]);
  REPEAT
    write('    LOOP COUNT (2,000..7,000) :   ');
    readln(LftLoops);
  UNTIL (LftLoops > 1999) AND (LftLoops < 7001);
  writeln;
  writeln;
  writeln('ENTER RIGHT ARRAY VARIABLES -');
  writeln;
  REPEAT
    write('    R. PIXEL DENSITY (8..40) :    ');
    readln(RtDotProb);
  UNTIL (RtDotProb IN [8..40]);
  REPEAT
    write('    LOOP COUNT (2,000..7,000) :   ');
    readln(RtLoops);
  UNTIL (RtLoops >1999) AND (RtLoops < 7001);
  writeln;
  Mem[$0000:$0417] := 32;                              { Sets NumLock ON }
  writeln;
  write('ALL ENTRIES CORRECT?  Y = YES,  N = REPEAT SCREEN:   ');
  readln(ch0);
UNTIL (ch0 = 'y') OR (ch0 = 'Y');
StoreConstantData;                                    { Store screen entries }
clrscr;
writeln;
writeln;
writeln;
writeln;
writeln;
write('                         Press  RETURN  to start.');
readln;
END;  { PROCEDURE SetUps }




{ --------------------------------------------- }
PROCEDURE StoreTestData;

BEGIN
writeln(F, 'ARRAY OFFSET:        ',OffSet);
writeln(F, 'L. # SYMBOLS:        ',LftSymbolCtr);
LftSymbolCtr := 0;
writeln(F, 'L. TARGETS:          ',LftTargSymCtr);
LftTargSymCtr := 0;                                    { Reset counters }
writeln(F, 'L. CORRECT:          ',LftRespCtr);
LftRespCtr := 0;
writeln(F, 'L. ERRORS:           ',LftErrCtr);
```

```pascal
LftErrCtr := 0;
writeln(F, 'R. # SYMBOLS:        ',RtSymbolCtr);
RtSymbolCtr := 0;
writeln(F, 'R. TARGETS:          ',RtTargSymCtr);
RtTargSymCtr := 0;
writeln(F, 'R. CORRECT:          ',RtRespCtr);
RtRespCtr := 0;
writeln(F, 'R. ERRORS:           ',RtErrCtr);
RtErrCtr := 0;
writeln;                                { Blank line to terminate file segment. }
END;  { PROCEDURE StoreTestData }

{ ------------------------------------------- }

PROCEDURE CloseFileF;

BEGIN
Close(F);
END;  { CloseFileF }

{ ------------------------------------------- }
PROCEDURE SetGrafix;

BEGIN
GrDriver := Detect;
InitGraph(GrDriver, GrMode, 'c:\tp\bgi');
{ --
NOTE: for production, set InitGraph to get BGI stuff in default
directory, where I will put it.
-- }
GrError := GraphResult;                              { GraphResult is in BGI }
IF GrError <> GrOK                                   { GrOK in BGI  }
THEN
    writeln('GRAPHICS ERROR: ', GraphErrorMsg(GrError));
END;  { PROCEDURE SetGrafix }
{ ------------------------------------------- }
PROCEDURE GrafixMsgs;
BEGIN
SetTextStyle(SmallFont, HorizDir, CharSize);
color := 15;                                            { white points }
nocolor := 0;                                    { black is the nocolor }
SetTextJustify(CenterText, CenterText);              { Messages to Screen }
OutTextXY((GetMaxX div 2), 15, 'Target Symbol is  ');
OutTextXY((GetMaxX div 2)+(TextWidth(TargSym)*8), 15, TargSym);
SetTextJustify(LeftText, TopText);
END;  { PROCEDURE GrafixMsgs }

{ ------------------------------------------- }
PROCEDURE Initialization;
BEGIN
chx  := 'a';                                        { Begin initialisation }
TimingCtr := 0;                                         { Start time @ 0! }
LftY := (GetMaxY div 2) - 50;             { both arrays on same horizontal }
RtY  := (GetMaxY div 2) - 50;
LftSymbolCtr := 0; RtSymbolCtr := 0;                    { total # symbols }
LftLoopCtr := 0; RtLoopCtr := 0;                 { Loops for symbol display }
LftTargSymCtr := 0; RtTargSymCtr := 0; TargSymCtr := 0;
LftRespCtr := 0; RtRespCtr := 0; RespCtr := 0;    { Correct response ctrs. }
LftErrCtr := 0; RtErrCtr := 0;                              { Error ctrs. }
LftTargSymOn := False; RtTargSymOn := False;       { TargSym on screen? }
RandSeed := 314759;                    { Same random number sequence for everyone. }
END;  { PROCEDURE Initialisation }
```

A-5

```
{ *********************************************************************** }
{ ------- NOW, BEGIN the MAIN program --------- }
BEGIN

REPEAT

  SetUps;                                          { get & set program variables }
  SetGrafix;                                              { Initialize graphics }
  Initialization;                                  { Initialise Variables, etc. }

{ -------- Start of main display loop, Exit on SessionCtr = 0 ------- }
REPEAT
{ ------- Set up array position from OffSet -- }
      OffSet := OffSetArray[OffSetArrayIdx];
      LftX := ((GetMaxX div 2) - 100) - (OffSet div 2);     { array OffSet }
      RtX  := (GetMaxX div 2) + (OffSet div 2);
    { Write the TargSym cue every time. }
      GrafixMsgs;
    REPEAT
{ ------- Left Array ----------------------- }
      x0 := random(100) + LftX;                          { Image at LftX, LftY }
      y0 := random(100) + LftY;
      IF random(LftDotProb) = 0 THEN
      PutPixel(x0, y0, color)                             { Position pixel }
      ELSE
      PutPixel(x0,y0, nocolor);                              { Omit pixel }
      Inc(LftLoopCtr);                                 { Increment counter }

      IF (LftLoopCtr = LftLoops) THEN         { Ck. ctrs. for symbol display }
      BEGIN
      Inc(LftSymbolCtr);                              { Aha! another symbol }
      x1 := (random(80)+10) + LftX;                   { Set Symbol location }
      y1 := (random(80)+10) + LftY;                      { dots and position }
      sym0 := symbol[random(5)+1];
      OutTextXY(x1, y1, sym0);
        IF (sym0 = TargSym)
        THEN
          BEGIN
          LftTargSymOn := True;
          Inc(LftTargSymCtr);
          Inc(TargSymCtr)
          END
        ELSE
        LftTargSymOn := False;

        k0 := (LftLoops div 2);                         { Introduce variability }
        k1 := random(k0) - (k0 div 2);                               { OK }
        LftLoopCtr := k1;                     { Reset @ LftLoops +- LftLoops/4 }

      END; { IF LftLoopCtr }

{ ------- End  Left Array ------ }

{ ------- Right Array ---------- }

      x0 := random(100) + RtX;                           { Image at RtX, RtY }
      y0 := random(100) + RtY;
      IF random(RtDotProb) = 0 THEN
      PutPixel(x0, y0, color)                               { Put pixel }
      ELSE
```

```pascal
      PutPixel(x0,y0, nocolor);                                 { Or dont put it }
      Inc(RtLoopCtr);
      IF (RtLoopCtr = RtLoops) THEN              { Begin Check Right Loop Ctr }
      BEGIN
      Inc(RtSymbolCtr);                          { Yet another bloody symbol. }
      x1 := (random(80)+10) + RtX;
      y1 := (random(80)+10) + RtY;
      sym1 := symbol[random(5)+1];
      OutTextXY(x1, y1, sym1);
        IF (sym1 = TargSym)
        THEN
          BEGIN
          RtTargSymOn := True;
          Inc(RtTargSymCtr);
          Inc(TargSymCtr)
          END
        ELSE
        RtTargSymOn := False;

        k0 := (RtLoops div 2);                              { Adds variability }
        k1 := random(k0) - (k0 div 2);                                    { OK }
          RtLoopCtr := k1;                      { Reset @ RtLoops +- RtLoops/4 }
      END; { IF RtLoopCtr  }

{ ------- End Right Loop ------ }

      FOR ctr0 := 1 TO SlowDown DO                       { SlowDown multiplier  }
      FOR ctr1 := 1 TO 30 DO;                    { 100 microsec delay per unit. }

{ ----- Begin the keyboard monitor routine. ----- }
{ ----- All Keys other than '4' and '6' are ignored. ----- }
      IF KeyPressed THEN
      BEGIN
      ch0 := ReadKey;
        IF (ch0 = '4') AND (LftTargSymOn = False) THEN
        BEGIN
          LftErrCtr := LftErrCtr + 1;
          Beeper
        END;
        IF (ch0 = '4') AND LftTargSymOn THEN
        BEGIN
          Inc(LftRespCtr);
          Inc(RespCtr);
          LftTargSymOn := False
        END; { IF }
        IF (ch0 = '6') AND (RtTargSymOn = False) THEN
        BEGIN
          RtErrCtr := RtErrCtr + 1;
          Beeper
        END;
        IF (ch0 = '6') AND RtTargSymOn THEN
        BEGIN
          Inc(RtRespCtr);
          Inc(RespCtr);
          RtTargSymOn := False
        END;   { IF }
      END; { IF KeyPressed }
{ ------- OK, End KeyPad Check ------ }

      Inc(TimingCtr);
    UNTIL (TimingCtr = TimeRef);
      Dec(SessionCtr);                                              { Completed? }
```

A-7

```
        Inc(OffSetArrayIdx);                                              { Next! }
        TimingCtr := 0;                                          { Reset clock }
        StoreTestData;                                            { Store data }
        ClearDevice;                                          { Clear display }

UNTIL SessionCtr = 0;                                            { Last epoch? }
CloseFileF;                                               { Close output file }
CloseGraph;                                                         { Must do! }
clrscr;
writeln;
k0 := (RespCtr * 100) DIV TargSymCtr;              { Calculate S's "score." }
writeln;
writeln('                    SUBJECT''S SCORE WAS   ',K0);
writeln;
writeln('                         SESSION COMPLETED');
write('          ·   HIT X TO EXIT, ANYTHING ELSE TO REPEAT.');
readln(ch0);
UNTIL (ch0 = 'x') OR (ch0 = 'X');

END.
```