A versatile, user-friendly tachistoscope for the Macintosh

JOSHUA M. DOENIAS, SUSAN E. LANGLAND, and DANIEL REISBERG
Reed College, Portland, Oregon

A wide range of experimental procedures require precisely timed stimulus presentations, and then record timed responses from subjects. In many of these procedures, stimulus presentations must be very brief (i.e., tachistoscopic), and followed by a masking stimulus. TScope, a new application for the Macintosh, is designed to run virtually any of these procedures. TScope and its companion program, ConfigureTScope, are designed with an emphasis on versatility, ease-of-use, and, consequently, ease of instruction. TScope is suitable for both classroom and research use. The present article provides a description of TScope's capabilities and also provides several examples of how this application can be used.

Many of psychology's experimental procedures require precisely timed stimulus presentations and record timed responses from subjects. These procedures have typically been run with tachistoscopes, but increasingly have more conveniently been run on microcomputers. As a result, a range of software is available for running these procedures, either for research purposes or for classroom demonstrations. However, much of this software has been designed to implement specific experimental tasks. Such canned procedures are easy to use, but at the cost of versatility.

The present article describes a tachistoscope program versatile enough to run a broad variety of procedures, including many already in the literature. The software is designed to allow variations on these basic procedures and the creation of new procedures. Crucially, the software is also designed to be extremely easy to use and correspondingly easy to learn. In particular, we wanted a program that could be explained in a few minutes to anyone familiar with the terminology of tachistoscopic presentation ("mask," "fixation point," and so on). For users not familiar with these terms (e.g., students in entry-level courses), we wanted a program that could comfortably be explained in less than an hour (i.e., in less than a single class period). TScope and its accompanying program, ConfigureTScope, are designed to meet these goals. In this paper, we describe how to use TScope, and, in the process, we describe the options and flexibility that TScope allows. In addition, we present a number of examples, to illustrate the range of procedures for which TScope is useful.

This software was prepared in the Reed College Software Development Laboratory. J. Doenias is currently at NeXT Computer, Inc. Address correspondence to D. Reisberg, Department of Psychology, Reed College, Portland, OR 97202 (or reisberg@reed.edu).

Note: One or more of the authors or their institution has a financial interest in the software described in this paper.—*Editor*

CONFIGURETScope

The ConfigureTScope application is used to set parameters that remain constant for all trials within an experiment. If a parameter is to vary from trial to trial, ConfigureTScope allows the user to specify that this parameter will be read, for each trial, from the stimulus file. When ConfigureTScope is run, the user is presented with a window containing a list of the possible steps in each trial. Each step is accompanied by a pop-up menu, used for specifying how that step should proceed.

ConfigureTScope allows the user to specify 11 parameters for each trial. The parameters, and the available options, are listed below:

1. What will initiate the trial? ConfigureTScope allows a trial to be initiated by a buttonpress by the subject, or by a timer. (A setting of zero on the timer is possible, causing each trial to follow immediately after the preceding trial.)
2. Will a fixation point be displayed before each stimulus is shown? The user can select no fixation point; a single dot as fixation point, located in the center of the region in which the stimulus will appear; or a picture frame, enclosing the region in which the stimulus will appear.
3. If a fixation point is to be displayed, its duration can be controlled either by a timer or by a response from the subject (i.e., the fixation point will remain in view until the subject makes a response).
4. The stimulus is presented immediately after the fixation point is removed. The stimulus duration, however, is specified by the user. Duration can be controlled by a timer (constant from trial to trial) or read from the stimulus file (and thus variable from trial to trial), or it can be contingent on the subject's response.
5. An experimental procedure may require a keyboard response to the stimulus from the subject; ConfigureTScope therefore asks a yes/no question about whether this response should be recorded. ConfigureTScope also asks

whether the actual key pressed should be recorded, or the response latency, or both.

6. Should the stimulus be followed by a mask? The user can select no mask, a flash mask, or a pattern mask. Alternatively, the user can indicate that the mask pattern will be specified in the stimulus file (allowing the mask to vary from trial to trial). If a pattern mask is selected, the program offers options for what this pattern should be.

7. If a mask is to be shown, what should its duration be? Mask duration can be controlled either by a timer or by the subject's response.

8. Many procedures require that subjects make a keyboard response or answer a question after the stimulus has been removed. In some cases, this requires that the computer's screen be blank for some period after the initial stimulus and before the probe or question appears. To allow this option, ConfigureTScope allows the user to specify whether a blank screen should be displayed after the stimulus.

9. If a blank screen is to be displayed, what should its duration be? As with other steps, this duration can be controlled either by a timer or by the subject's response.

10. Finally, will some further stimulus, probe, or question about the initial stimulus be presented? A yes/no decision is required here.

11. If questions are to be asked at the end of each trial, should the subject's responses to these questions be collected? If so, should the program record the responses, the response latencies, or both?

THE STIMULUS FILE

Many parameters of a procedure will be constant from trial to trial; these are set by the ConfigureTScope program. Other parameters, however, including the stimuli themselves, will vary from trial to trial. These must be specified in a *stimulus file*. Any text editor or word processor may be used to create a stimulus file, with the constraint that the stimulus file must be saved in text-only format.

The TScope program allows four parameters to vary from trial to trial:

1. The *stimulus* to be presented on that trial. (This can be any text; it is displayed in the Geneva font.)

2. *Duration* of the stimulus presentation for that trial. (Stimulus duration, like all presentations in TScope, is synchronized with the vertical retrace.)

3. The form of the *mask* for that trial, with choices among a flash mask, a mask of number signs ("#####"), or a mask of random characters.

4. The *question* field for that trial.

The user can specify (or omit) any combination of these parameters from the stimulus file. Parameters not specified will be read from the configuration file and thus will be the same for all trials in the procedure. For example, imagine a procedure in which subjects are shown pairs of stimuli and must judge whether the stimuli within each pair are synonyms or not. Thus, in one trial, the subject

might see the word *stone* and then the word *rock*, and should answer "same." In the next trial, the subject might see *apple* and then *tree*, and should answer "different." For this procedure, the stimulus duration and the mask would be the same from trial to trial, but each trial would have a unique stimulus, followed by a unique question or probe. Thus, the stimulus file would identify *stone* and *rock* as the stimulus and question (respectively) for one trial; *apple* and *tree* would be the stimulus and question for the next trial. Note how the stimulus and question fields are being used here—as two text fields, each presenting a stimulus word. Thus there is no requirement that the stimulus field present a stimulus to be judged, or that the question field present a question about that stimulus. For this procedure, neither duration nor mask would be specified in the stimulus file; these parameters would therefore be read from the configuration file.

As a different example, consider a procedure in which subjects are being trained in solving arithmetic problems; the procedure requires that subjects gradually increase their speed. Specifically, subjects are given 3 sec to solve the first problem, 2.5 sec to solve the second, 2 sec to solve the third, and so on. In this case, the duration field would be used to control the pace of the procedure, and the stimulus field would contain the arithmetic problems (4×28 , 5×19 , etc.). No mask or question would be specified in the stimulus file, and therefore these parameters would be read from the configuration file (and so would be constant from trial to trial).

Each line within the stimulus file provides parameters to be used for one trial. In many cases, the first line of the stimulus file will provide information to be used in the first trial; the second line, information for the second trial; and so on. However, the TScope program provides considerable flexibility in this sequencing.

THE TSCOPE APPLICATION

In using the TScope program, the user must first specify a configuration file and a stimulus file. The user then selects "Run trial." The computer screen is then blanked, and the message "Hit any key to begin trial" is displayed. Once the subject has pressed a key, the sequence of trials is run. The format of each trial is governed by the configuration file; the number of trials overall is determined by the number of lines in the stimulus file (since each line provides information for one trial).

TScope allows the user to present the trials in a sequence different from that in the stimulus file itself. Several options for changing this sequence are available; moreover, different options can be used in different phases of an experimental procedure. A concrete example will make this clear. An investigator wishes to examine whether response times in a lexical decision task depend on the semantic category of the probed word. Thus, in the experimental trials, one fourth of the stimuli to be presented might be nonwords, one fourth might be names of fruits, one fourth might be names of vehicles, and one fourth, names of

colors. Let us say that 5 words are included in each category, for a total of 20 words overall. Finally, the researcher might want an additional 5 words to be shown at the beginning of the procedure, to acquaint the subject with the task.

In preparing the stimulus file for this procedure, the researcher might find it convenient first to type in the five practice words, then the five nonwords, then the five fruits, the five vehicles, and the five colors. However, for most purposes, it would not be appropriate to present the words in this sequence (given concerns about fatigue, etc.). The researcher could manually rearrange these sequences by retyping the stimulus list, and so mix together the categories of words. But this would be cumbersome, and it would have to be redone for each subject (if each were to get a different random sequence). TScope provides an easier alternative.

Within TScope, the researcher would first specify the configuration and stimulus files. The user would then select "Change sequences" from the Sequences menu. This calls up a dialog box that allows the researcher to specify whether all of the stimuli, as a group, are to be resequenced, or whether individual sections of the stimuli will be sequenced, section by section. In the immediate example, the user might wish to randomize the sequence of practice words, but does not wish to interweave the practice words with the test stimuli themselves. However, the test stimuli themselves do need to be randomized. Hence, in this case, there are two sections (practice and test) to be sequenced separately. This division between sections will be invisible to the subject; it only matters for the program's operation. (For illustration, both the stimulus list and TScope's treatment of the list are shown in Figure 1.)

The program then requests further information about the sequencing—how many words there are in each sec-

tion, and how that section is to be sequenced. In the example, the user would indicate that there are five words (the practice words) in the first section; the user could then select the option of randomly sequencing these words. Thus, because of the sectioning, the first five words presented in the actual procedure will always be the first five words in the stimulus file; the sequence within this five will be randomly determined for each subject, however.

For the test stimuli, the researcher might prefer a sequence of presentations that is sensitive to the fact that the experiment includes four categories of strings (nonwords, fruits, vehicles, colors). In this case, the user would select "scramble by categories." This brings up a dialog box that allows the user to specify that this section of the procedure includes four categories of stimuli. TScope is now led to interpret the stimulus file as shown in the lower right corner of Figure 1.

One option available at this point would be to randomize "within categories." In this case, the program would first present all the words in the first category (in this example, the five nonwords), but with the sequence within this category randomly determined. Once all the nonwords had been presented, the program would do the same for the second category (i.e., the fruits)—present these in some random sequence. The program would then do the same for the vehicles and colors.

A different option would be to "randomize categories." In this case, the program would first present the words in the first row of the table in Figure 1 (i.e., *pirt*, *apple*, *truck*, *red*), with the sequence within this row randomly determined. Once all the words in this row had been presented, the program would do the same for the words in the second row in the table (*tappek*, *pear*, . . .), and so on for each subsequent row. In this way, the categories themselves would be randomly sequenced, and each group

Figure 1. The sequence of presentations can be configured in a variety of ways. See text for details.

of four words within the presentation would include one word from each category. However, the first fruit to be presented would always be the first fruit in the stimulus file (*apple*); the second fruit to be presented would be the second in the file (*pear*); and so forth.

The user also can select *both* of these resequencing options. In this case, the sequence will be the one we desire for this application: random, with the constraint that each group of four trials include one trial from each category.

Finally, if data are to be recorded from an experimental procedure, these resequencing options could easily lead a researcher to lose track of which responses were made to which stimuli. To avoid this, TScope provides two options for how the data can be recorded. First, the data can simply be recorded in the sequence of the actual trial presentations. Alternatively, the data can be recorded in a sequence corresponding to the sequence in the stimulus file, rather than the presentation sequence. To continue the example illustrated in Figure 1, this would cause the first five trials recorded in the data file to be the five practice trials; the next five trials recorded would be those with the nonwords, and so on. In short, the sequencing options within TScope will scramble these categories together in presentation, but keep them unscrambled in the data record.

At the end of an experimental procedure, TScope records the data in a file named by the user. The data are stored in tabular form in a text-only file. This file can then be read by a variety of analysis or charting programs.

EXAMPLES

The principal advantage of TScope lies in its versatility, for the program will support a wide range of procedures. In this and subsequent sections, we briefly sketch how the program might be configured to run these procedures.

Word-Superiority Effect

The word-superiority effect (Reicher, 1969; Wheeler, 1970) consists of an advantage in the visual processing of well-formed visual strings (words or orthographically regular nonwords) in comparison with isolated letters or irregular nonwords. To minimize the contribution of guessing, this advantage is often assessed with a forced-choice procedure. For example, subjects might be shown the string CARP and then asked whether a *p* or a *t* appeared in the display; subjects might then be shown the string XJFP, and likewise be asked whether a *p* or a *t* had appeared. In procedures like this, word superiority emerges as a greater likelihood of correct choice in the former case.

With TScope, the configuration file would be used to specify all parameters except for the stimulus and question text. That is, duration, mask, and so on would all be the same for all trials. The duration would presumably be brief (33 or 48 msec); a poststimulus pattern mask would presumably be employed. The configuration file would also specify that no response was required to the

stimulus itself, but that a response was required to the question when it was presented. The stimuli for this procedure, and the pair of test letters for each trial, would be listed in the stimulus file. For example, CARP might be the stimulus for one trial, and *p* or *t* the corresponding question; XJFP might be the stimulus for another trial, and *p* or *t* its question.

It might be convenient in this case first to type the practice words into the stimulus file, then the test words, then the nonwords. In running the trials, the practice words could be randomly sequenced; the words and nonwords would then be interwoven via "randomize categories."

Memory Scanning

A memory-scanning procedure (e.g., as in Sternberg, 1969, 1975) would employ a set-up similar to that in the first example. In this case, however, stimulus durations would be relatively long (3–4 sec), and no mask would be employed. The configuration file would specify that no response was required to the stimulus itself but that a response was required to the question. The stimulus file would contain, for each trial, a string of letters (in the stimulus field) as the set to be scanned, and a probe letter (in the question field).

Serial Position Effect

Many procedures do not require brief stimulus exposures, but do require that the exposures be precisely timed. TScope is also useful for these procedures. For example, in a procedure on list learning (see, e.g., Glanzer & Cunitz, 1966; Murdock, 1962), TScope could be used to present a list of words for 2 sec each, followed by a "RECALL THE LIST" instruction, displayed for 10 sec. In resequencing the list, the user would use "sections" to make certain that the "Recall" instruction was the last presentation in the series, while at the same time allowing random sequencing of the stimulus words.

Sentence Verification

In a sentence-verification procedure (see, e.g., Collins & Quillian, 1969; Rosch, Simpson, & Miller, 1976), subjects are presented with a series of sentences and must indicate whether each is true or false. In this case, the sentences would be presented within the stimulus field; nothing would appear in the question field. Stimulus duration would be controlled by the subject's response (i.e., the sentences would remain in view until the subject pressed a key). No mask would be employed.

Priming

In a classic demonstration of repetition priming, Posner (1978) used a warning signal in each trial to prime subjects for upcoming material. For example, the subjects' task was to decide whether two letters were the same or not (answering "same" for the pair "A A" and "different" for the pair "A B"). Decision times were shorter if the warning signal for that trial matched the to-be-judged

letter (e.g., the warning signal was an "A"); apparently, the warning signal had primed the subject for the upcoming stimulus.

To run a procedure of this sort, TScope's stimulus field would be used, not for the test stimulus itself, but for the warning signal (the prime); the question field would then be used for the test pair. No response would be required to the stimulus (in this case, to the prime), but responses to the question would be recorded (with one key on the keyboard designated as a "same" response, and one as a "different" response). The interval between the prime and the test pair can easily be manipulated by using ConfigureTScope to control the duration of the blank screen inserted between the stimulus and the question.

SYSTEM REQUIREMENTS AND AVAILABILITY

TScope and ConfigureTScope are written in the Rascal Programming Environment and require at least a Macintosh Plus with 1 MB of memory. Both programs require System 6.0 or higher and are fully compatible with MultiFinder. If TScope is run with a color monitor, it will automatically set the system to monochrome. (The authors are working on an upgrade of TScope to support color displays. The upgrade will also allow small PICT files to be used as stimuli, in addition to alphanumeric stimuli, and will allow control over display font.) TScope and Con-

figureTScope are available directly from the third author at \$20.00 for single copies. (The upgrade will be free to all purchasers of the current version.) A discount is available for purchase of multiple copies.

REFERENCES

- COLLINS, A. M., & QUILLIAN, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning & Verbal Behavior*, **8**, 240-247.
- GLANZER, M., & CUNITZ, A. R. (1966). Two storage mechanisms in free recall. *Journal of Verbal Learning and Verbal Behavior*, **5**, 351-360.
- MURDOCK, B. B., JR. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, **64**, 482-488.
- POSNER, M. I. (1978). *Chronometric explorations of mind*. Hillsdale, NJ: Erlbaum.
- REICHER, G. (1969). Perceptual recognition as a function of the meaningfulness of the stimulus material. *Journal of Experimental Psychology*, **81**, 275-280.
- ROSCH, E., SIMPSON, C., & MILLER, R. S. (1976). Structural bases of typicality effects. *Journal of Experimental Psychology: Human Perception & Performance*, **2**, 491-502.
- STERNBERG, S. (1969). Memory-scanning: Mental processes revealed by reaction-time experiments. *American Scientist*, **57**, 421-457.
- STERNBERG, S. (1975). Memory-scanning: New findings and current controversies. *Quarterly Journal of Experimental Psychology*, **27**, 1-32.
- WHEELER, D. (1970). Processes in word recognition. *Cognitive Psychology*, **1**, 59-85.

(Manuscript received March 19, 1991;
revision accepted for publication August 26, 1991.)