METHODS & DESIGNS

Eye-movement data: Cumulative fixation time and cluster analysis

C. R. LATIMER

University of Sydney, Sydney, New South Wales, Australia

This paper describes procedures for the visual presentation and cluster analysis of eye-movement data. Methods for two- and three-dimensional representation of cumulative fixation time (CFT) and ways of enhancing the peaks in CFT distributions are outlined and illustrated by reference to examples from eye-movement studies of cognitive processes. CFT distributions may also be partitioned using the *k means* clustering technique (MacQueen, 1967), and applications of variants of this technique to eye-movement data are discussed. Cluster analyses such as *k means* require the user to make initial estimates of the number and value of the means. One classification procedure (Wallace & Boulton, 1968a, 1968b), based on information theory, avoids these initial assumptions. This procedure is applied to a CFT distribution and has its solution compared with that of *k means* for the same distribution. Finally, programs that implement these procedures on Macintosh computers are listed and offered on floppy disk.

Arguments and evidence for the reliability and validity of eye fixations and fixation time as indices of cognitive processes are well documented (Just & Carpenter, 1976; Tichomirov & Posnyanskaya, 1966; Winikoff, 1967). Furthermore, examples of the use of these indices in experimental investigations of cognition are many and varied. Mackworth and Morandi (1967) made comparisons between visual fixations on, and verbal estimates of the relative importance of, regions within photographs. These authors found that the regions that were rated highly for informativeness produced the highest fixation frequencies. Baker and Loeb (1973) found appreciable correlations between ratings of the importance of sections of geometric forms and durations of fixations on those sections. In a developmental context, Nodine and Steuerle (1973) studied the eye movements and fixations of kindergarten children during differentiation of matched and unmatched letter pairs. The letters were differentiated by the presence or absence of distinctive features within a 2×2 grid, and, with each letter subtending an angle of 14° and the letters being 24° apart, it was possible to record fixations

on individual letter segments. In all the above examples, strong reliance was placed on relationships between (1) measures of fixation frequency and cumulative fixation time on particular regions of stimuli and (2) underlying cognitive processes.

This paper reports techniques for visual presentation and quantitative analysis of these particular measures of eye-movement behavior. As Scinto and Barnette (1986) pointed out, progress and innovation in developing analytic techniques for the assessment of eye-movement data, particularly of overall distributions of fixations and fixation time, have been slow in comparison with advances in related domains, such as eye-movement recording methods. The techniques described here are particularly useful when an experimenter wants some *quantitative* method of discovering (without immediate recourse to the stimuli themselves) what stimuli or parts of a stimulus have received most attention during inspection by subjects. These procedures provide a basis for ordering stimuli or stimulus parts in terms of salience, necessity and sufficiency for recognition, importance, or, indeed, any descriptive dimension that may be indexed by eye fixation. Ultimately, of course, interpretations of eye-movement data must be made with respect to the stimuli, but these methods are appropriate when, in the first instance, an experimenter wants to assess the relative importance of stimuli or stimulus parts directly from the eye-movement data. It should also be noted that procedures described here do not take into account dynamic aspects of eye-movement data, such as order of scan or saccade length. For methods that consider these aspects, the reader is referred to Latimer (1978) and Scinto and Barnette (1986).

This research was supported by an Australian Commonwealth Postgraduate Scholarship and by University of Sydney Research Grants. The eye-movement data were collected using facilities at the Department of Artificial Intelligence, University of Edinburgh. I express gratitude to my supervisors, J. P. Sutcliffe and J. A. M. Howe, for their help and encouragement; to A. J. Marley and G. Oliphant for many helpful discussions on these matters; and to D. Boulton for running the program SNOB on my data. I also acknowledge the contributions of George McConkie and another anonymous reviewer, whose discerning critical comments helped to clarify a number of issues in the paper. The author's address is Department of Psychology, University of Sydney, New South Wales 2006, Australia.

I begin by describing some fairly simple methods for topographical representation of an overall distribution of cumulative fixation time (CFT) and relating it to stimuli, before going on to present more quantitative methods of assessing such distributions, such as cluster analysis. More elaborate descriptions and examples of all these methods can be found in Latimer (1978).

CUMULATIVE FIXATION TIME

A CFT plot is simply a topographical representation of the cumulative fixation times made by a subject while inspecting a stimulus. Such plots also can be cumulative across stimuli and/or subjects. For example, a subject, during inspection of a stimulus on one trial or a set of stimuli over many trials, makes five fixations on a particular location on the screen, and the fixation times are, in seconds, .23, .50, .67, .30, and .32. In a CFT plot, this screen location would be given a value of 2.02 (the sum of the fixation times). The CFT measure is close to what Just and Carpenter (1976) referred to as "location and duration of gaze," although these authors did not accumulate fixation time to the extent reported here.

By way of illustration, consider the following very simple experiment. A subject sits in front of a large screen subtending 57° high and 39° wide, and is asked to search for an oblique line subtending 27.32°. The line can appear randomly in one of the following positions: center, top left, top right, bottom left, or bottom right. The subject is given a search start point chosen randomly from these same positions. On each trial, the stimulus line is presented under conditions of high illumination and low contrast, which create a form of tunnel vision; the subject can see clearly only within a very circumscribed region around fixation. The effect is similar to looking at the screen through a narrow pipe allowing a circular area of approximately 12° of visual angle to be viewed clearly at each fixation. The subject is allowed 15 sec to find the stimulus line, and eye movements and fixations are recorded by electrooculography during search. This experiment was carried out as a pilot study prior to further experimentation on feature extraction and feature weighting in uppercase letter recognition under similar tunnel-vision-producing conditions. Some of the relevant main objectives of the pilot study were to achieve estimates of the degree of tunnel vision engendered by high illumination and low contrast; to study the general form of the CFT distribution across subjects; to establish that, under the tunnel-vision conditions, subjects do not scan the screen randomly, but that fixations and CFT are influenced by stimulus dimensions; and to look for individual differences in CFT patterns and compare individual CFT with accumulated CFT across all subjects.

Figure 1 is a two-dimensional representation of CFT for one subject across 25 trials, and, for convenience, the stimulus line has been superimposed on the five screen positions in which it could appear during the experiment.

The number border provides an index of screen locations in inches, and the values on the screen are CFT values (t values) for each screen location. In order to conform to printout limitations, each t value has been multiplied by 10 and truncated to two figures. For instance, in the example above, the t value 2.02 would become 20. A t value of 5.23 would print as 52, and so on.

In this way, a subject's fixations and fixation times can be related in a very direct manner to the stimuli; areas of the screen given a lot of attention can be readily seen from this overall distribution. For example, it is fairly clear from Figure 1 that this subject's fixations and fixation time have not been distributed randomly across the screen, but have been concentrated mainly on areas of the screen where the stimulus line could appear. Fixations and fixation time also accumulate in the screen center, partly because of the convergence of positions for two line stimuli in this area and partly because of the screen center being a "crossroad" position between stimuli placed at screen corners. The small concentration of CFT to the right of center may also indicate a similar "crossroad" between the top-right and bottom-right positions. Thus, an experimenter can begin to build up a picture of exactly where attention has been assigned on the screen, and which stimuli or stimulus regions have attracted this attention.

The algorithm for plotting CFT, which is fairly simple, is embodied in a computer program, CFT_Analyzer (see Appendix A for program listing and Appendix B for documentation). The program reads a text file of x - y coordinates and fixation times, and then (1) converts the x - y coordinates to row and column coordinates, (2) accumulates the fixation times for each location, and (3) prints out the truncated t values in their screen locations. The code for this program was originally written in FORTRAN, but a version in True BASIC is provided in Appendix B.¹

Three-Dimensional CFT

There are various methods and some programs for detecting and depicting the peaks and troughs in digital arrays, including the drawing of contour lines (e.g., Johnston & Rosenfeld, 1975). The method used here is to create a three-dimensional representation of the CFT distribution as in Figure 2. To obtain the values for this representation, the original screen matrix of 79×52 locations has been reduced to a matrix of 13×13 locations. For example, t values with row coordinates 1-6 and column coordinates 1-4 sum to provide the value for row 1, column 1 of the new 13×13 matrix; t values with row coordinates 7-12 and column coordinates 5-8 sum to make the value of row 2, column 1; and so on. Thus, the original 79×52 representation of CFT is coarsened to a 13×13 representation. With the 13×13 values calculated, any three-dimensional plotting routine can be applied to give the effects in Figure 2. The particular routine used in this case was written in True BASIC and

Figure 1. A two-dimensional CFT distribution for 1 subject across 25 trials attempting to detect an oblique target line that has been plotted in its five possible positions.

Figure 2. A three-dimensional representation of the distribution shown in Figure 1 after reduction from 79×52 to 13×13 rows and columns. Viewing angle is from the bottom left-hand corner of the stimulus screen (Figure 1). View coordinates: $x = -30$; $y = -60$; $z = 60$; Distance = 50; Z_Divisor = 4.

allows viewing of the distribution from different angles, heights, and distances. The particular view in Figure 2 is from the bottom left-hand corner of the stimulus screen (Figure 1). Subroutine Three_D_Plotter (Appendix A), which prompts the user for viewing coordinates that can range right around the compass, is used to produce these three-dimensional images. No part of the three-dimensional surface need be hidden. Areas obscured from one set of viewing coordinates can be seen easily from other viewing positions. Figure 4 (view from bottom right), Figure 5 (view from top right), and Figure 8 (view from bottom right) illustrate the effects of changing coordinates. The views in Figures 5 and 8 are also from higher elevations.

The degree of coarsening used for three-dimensional plotting is important and could, in theory, range all the way from the original CFT matrix dimensions to a one-cell matrix with t values averaged overall. Too little coarsening, in its failure to summarize, can result in a confusing three-dimensional plot in which general areas of high CFT are masked. Too much coarsening tends to mask secondary and lower peaks in the CFT distribution, and, unfortunately, there are no hard and fast rules for deciding exactly what degree of coarsening should be used for every possible data set. Nonetheless, one way of proceeding is to start as close to the original CFT matrix dimensions as computer and program memory will allow, and, having established that the highest peaks in the two-dimensional distribution are in fact represented by the highest columns, continue coarsening and ensuring that greater and lesser peaks retain their representation in three dimensions. Another way of proceeding is to start at the coarse end of the spectrum and ensure that the highest peaks in the two-dimensional distribution all fall roughly in the center of three-dimensional columns. With further refinement of the matrix, these higher peaks should retain their positions on the most prominent columns, lesser

peaks should begin to take up position on columns at the next highest level, and so on. The difficult question with either of these procedures is, of course, when to stop the coarsening or the refinement. There is no simple answer to this question, but it should be remembered that the three-dimensional plots are simply a means of summarizing and highlighting information already existing in the two-dimensional CFT matrices and no more. It is ultimately up to the user of the procedure to decide the level of coarsening at which important information in the two-dimensional distribution is being lost in three dimensions.

A comparison of Figures 1 and 2 is instructive. The highest columns in Figure 2 correspond very closely to and highlight the clusters of high t values evident in Figure 1. The main peaks in Figure 1 are located on the highest columns of Figure 2, and, given the questions being addressed in this particular pilot study, it is doubtful whether further coarsening or refinement would provide additional important information. Clearly, the decision to coarsen or refine depends on the specificity of hypotheses under test in an experiment. As Just and Carpenter (1976, p. 470) pointed out, "The appropriate unit of analysis in relating eye fixations to cognitive processes depends on the theory motivating the analysis." Perhaps the same direction could be taken in the use of the coarsening procedure. If, for example, it is crucial for some theory under test that attention is concentrated on stimuli or stimulus parts on one screen location and not on stimuli or stimulus parts on some closely adjacent screen location, it may be necessary to retain most of the fine detail of the two-dimensional representation in three dimensions. The Baker and Loeb (1973) study, in which hypotheses concerning fixations and fixation time on specific and adjacent regions of stimuli were under test, is a good example of the sort of study for which close-grained three-dimensional representation would be required. In other analyses, such as our pilot study example, a more general indication and a much coarser three-dimensional representation of the CFT distribution would suffice.

CFT Across Subjects and Stimuli

CFT can also be accumulated over subjects and stimuli to reveal regions of the screen or stimuli that attract most attention overall. This is an important aspect of the procedure in that it allows for accumulation of fixation time over many stimuli, trials, and subjects, and thus increases the reliability of this eye-movement measure. Figures 3 and 4 are two- and three-dimensional CFT plots accumulated over 12 subjects $\times$ 25 trials. The stimulus line has been plotted in its five possible screen positions.

Here again, Figures 3 and 4 reveal the areas of high CFT across subjects and, together with individual CFT distributions, provide the basis for comparisons between subjects and between any particular subject's CFT distribution and the general CFT distribution. For example, there were no marked individual differences in the pilot study subjects' CFT distributions, and this is to some ex-

Figure 3. A two-dimensional cumulative CFT distribution across 12 subjects and 25 trials attempting to detect an oblique target line that has been plotted in its five possible positions. Circles mark the highest isolated peaks in the distribution.

Figure 4. A three-dimensional representation of the distribution shown in Figure 3 after reduction from 79×52 to 13×13 rows and columns. Viewing angle is from the bottom right-hand corner of the stimulus screen (Figure 3). View coordinates: $x = 30$; $y = -60$; $z = 60$; Distance = 50; $Z_Divisor = 30$.

tent indicated by the uniform location of peaks and global similarity of the CFT distributions depicted in Figures 1–4. The accumulation of CFT will have uses in other eye-movement studies of cognition. In studies of reading processes, for example, an experimenter wanting to compare the accumulation of fixation time across subjects and trials on different prose passages could employ this method of representing the CFT distributions.

CFT Thresholding

Peaks in a CFT distribution can be enhanced by successive removal of the lower t values. This is achieved by setting thresholds below which t values are not printed. The effect is like a morning mist rising in the valleys and leaving only the highest peaks in view. Figures 5 and 6 depict the distribution shown in Figure 3 with t values less than 9 removed, and t values less than 18 have been removed in Figures 7 and 8. Subroutine Thresholder (see Appendix A) can execute this thresholding.

Thresholding can provide a quick way of magnifying the regions of the stimulus or screen receiving the most attention, but, again, as in decisions relating to the degree of coarsening and refinement for three-dimensional plotting, the levels at which thresholds are set must depend on the specificity of the theory and hypotheses under test. In our pilot study example, it is sufficient to note the rough location and area of the peaks in the CFT distribution; however, in a study like that of Nodine and Steuerle (1973), where it was predicted that the most distinctive features of comparison-letter pairs should receive most attention during differentiation, thresholding could and should be carried out until the highest CFT peaks are shown to be located (or not located) over the distinctive features.

Summary Descriptive Statistics

In addition to the above measures, a set of summary descriptive statistics can be computed (Subroutine Summary_Stats in Appendix A) for each CFT distribu-

tion. These are the sums, means, maximum and minimum values, and standard deviations of the t values. Such descriptive statistics allow for a numerical comparison of subjects and distributions where these are thought necessary.

Interpreting the CFT Distributions

In summarizing the relevant main objectives of the pilot study, it is fairly clear from the two- and three-dimensional plots and the thresholding that (1) the main peaks in the CFT distributions lie, as would be expected, at locations very near the stimulus line in its five possible positions; (2) fixations and fixation time are not distributed randomly, but are determined by stimulus configuration in that they concentrate in the areas where the line was presented; and (3) although not all presented here, individual CFT distributions did not differ significantly in general form or location of peaks, and were for the most part reduced versions of the overall CFT distribution depicted in Figures 3 and 4. Finally, tunnel vision was actually measured by noting the length of a subject's final saccade onto the stimulus line prior to detection. These saccade lengths were averaged across trials and, in giving the approximate distance of fixation from the stimulus lines just prior to detection, gave an estimate of the radius of tunnel vision. This radial estimate averaged at around 5° to 6° or 7 to 8 in., indicating an area of clear vision of diameter 10° to 12° or 14 to 16 in.

Clearly, in a more sophisticated experiment designed to test precise hypotheses about which stimuli or parts of a stimulus should receive most and least attention, these procedures for calculation and representation of CFT could be used in tests of such hypotheses.

CLUSTER ANALYSIS

Some may argue that the foregoing methods involve a fair amount of subjectivity. For example, the decision as

Figure 5. A three-dimensional representation of the Figure 4 example of enhancing CFT higher peaks by thresholding. Viewing angle is from the top right-hand corner of the stimulus screen (Figure 6). View coordinates: $x = 30$; $y = 60$; $z = 200$; Distance = 120; $Z_Divisor = 15$.

Figure 6. An example of thresholding to enhance the higher peaks in a CFT distribution. In the CFT distribution presented in Figure 3, t values less than 9 have been deleted.

Figure 7. Further enhancement of the higher peaks in the CFT distribution shown in Figure 3. In this case, t values less than 18 have been deleted.

Figure 8. A three-dimensional representation of the Figure 7 example of enhancing CFT higher peaks by thresholding. Viewing angle is from the bottom right-hand corner of the stimulus screen (Figure 7). View coordinates: $x=30$; $y=-60$; $z=200$; Distance = 120; $Z_Divisor = 15$.

to what degree of coarsening should be applied to the CFT distribution prior to three-dimensional plotting might favor some interpretations of the data and not others. The choice of thresholds and the levels at which the lower t values are deleted could have similar effects. In consequence, it may be thought that cluster analysis could provide a more objective assessment of the CFT distributions under consideration. Some commentators (Blackith & Reyment, 1971, p. 278) have suggested, however, that a perusal of the voluminous literature on cluster analysis very quickly brings one to the disquieting conclusion that there are not, and cannot be, hard and fast rules about what really constitutes a cluster. This sort of consideration should be borne in mind throughout the remainder of the discussion on cluster analysis.

What are Clusters and Cluster Means?

Before proceeding with descriptions of clustering techniques, it should be made clear exactly what clusters and cluster means are, and exactly what psychological significance they might have within a CFT distribution. Clusters, in the present context, are heavy concentrations of fixations and fixation time (hence attention) on particular areas of a screen and stimuli. If a CFT distribution can be partitioned into such clusters in some relatively objective manner, an experimenter should be able to point to such clusters and assert that stimuli or stimulus parts located under them have received significant attention by a subject or subjects. For example, in an experiment in which young subjects have been asked to judge pairs of patterns as the same or different, (e.g., Nodine & Steuerle, 1973; Vurpillot, 1968), and the subjects have done so successfully, an experimenter may argue that the clusters within the CFT distribution indicate the most distinguishing features or regions of the patterns. In similar experimental contexts, an investigator may, on the basis of a theory (the utility theory, Pashner, 1971; or the differential concept formation theory, Sutcliffe, 1986), predict the most differentiating features of patterns and

use clusters within the observed CFT distribution to test these predictions. The mean of a cluster in a CFT distribution is assumed to be the central point of an area of the screen, a stimulus or a stimulus region for which the fixations within the cluster were intended. Given these assumptions, it is necessary to find some clustering technique applicable to CFT distributions.

The k means Cluster Analysis

MacQueen (1967) provided a fairly simple and direct clustering technique called k means. When applied, for example, to the CFT distribution in Figure 3, the objects to be clustered are the locations within the distribution, and I shall refer to these as *triples*. Each object or location has three descriptors (hence *triple*): its row coordinate, its column coordinate, and its t value. The k means technique requires, as a first step, an estimate of the number of clusters in the distribution, together with estimates of the mean row and column coordinates and the mean t value of each cluster. With this step, a large subjective element necessarily enters the analysis. The experimenter must make a judgment about what areas in the CFT distribution *might* constitute separable concentrations of fixations. Some guidance is given by the three-dimensional representations and the thresholding procedure, which highlight the isolated peaks in a distribution, but there is no clear, objective procedure for making these initial estimates. In the case of the distribution shown in Figure 3, an initial estimate of six clusters is defensible inasmuch as the initial mean estimates of these clusters are six high and fairly isolated t values, or peaks in the CFT distribution with their row and column coordinates. These initial cluster means are circled in Figure 3 and are also set out in Table 1.

With the initial estimates settled, the remaining triples (t values in the 79×52 matrix, together with their row and column coordinates) are now considered in random order for membership within each cluster. The criterion of membership is the Euclidean distance from each cluster mean, where each randomly chosen triple is allocated to its nearest cluster. After a triple has been allocated, new mean row and column coordinates and total t value are computed from the triples already in the cluster and the new triple. The process of random assignment of triples proceeds until all triples in the CFT distribution have been allocated and the final mean positions ascertained. For those who would prefer a formal account of the computations required for this sort of k means cluster analysis,

Table 1
Initial Mean Estimates for k means Cluster Analysis

Mean	Row	Column	t value
1	22	14	28
2	14	37	25
3	28	29	58
4	35	39	26
5	54	40	49
6	54	23	30

this is set out in Appendix C. However, the following informal description of *k means* analysis and its options may suffice for many. Take, for example, a very simple case of a CFT distribution with three initial cluster means at the point where 10 triples have been assigned to Cluster 1, 5 triples assigned to Cluster 2, and 22 triples assigned to Cluster 3. This position is depicted in Figure 9.

MacQueen's *k means* algorithm. Let *r* stand for row coordinate, *c* for column coordinate, *p* for number of triples already assigned to a cluster, and *tt* for the sum of the *t* values of triples already assigned to a cluster. Let *t* stand for the fixation time of a triple being assigned to a cluster. Thus, for example, Cluster 2 has a *tt* value of 45, a row coordinate of 25, a mean column coordinate of 40, and five triples assigned to it. At this point, the newly selected triple (*t*15: *r*36: *c*25) is assigned to the closest cluster, Cluster 2. The new mean row coordinate is computed for Cluster 2 according to the following formula:

r:- ROW COORDINATE c:- COLUMN COORDINATE
t:- FIXATION TIME
p:- NUMBER OF ASSIGNED TRIPLES
tt:- CUMULATIVE FIXATION TIME

Figure 9. A schematic representation of the assignment of a triple in the *k means* cluster analysis (Type 1, MacQueen's algorithm). See text for details.

$$[1/p_{(\text{Cluster } 2)} + 1] * [p_{(\text{Cluster } 2)} * r_{(\text{Cluster } 2)} + r_{(\text{new triple})}],$$

which is $[1/(5+1)] * [5 * 25 + 36]$, which equals 26.833, or 27 when rounded to the nearest integer.

A new mean column coordinate is computed for Cluster 2 in a similar manner:

$$[1/p_{(\text{Cluster } 2)} + 1] * [p_{(\text{Cluster } 2)} * c_{(\text{Cluster } 2)} + c_{(\text{new triple})}],$$

which is $[1/(5+1)] * [5 * 40 + 25]$, which equals 37.499, or 38 when rounded to the nearest integer.

Thus, Cluster 2 has its mean relocated at row 27 and column 38 and, with six triples now assigned to it, has a cumulative fixation time of 60. The means tend to wander around within the distribution, but MacQueen's (1967) proof applied in the present context would suggest that, if the triples in the CFT distribution are actually generated by *k* different probability distributions, then the *k* estimates will asymptotically converge on the means of the distributions.

When applied to a CFT distribution, MacQueen's original algorithm can be modified in a variety of ways. I discuss only one of these possible modifications.

CFT weighting. Instead of using number of triples assigned to Cluster 2 plus 1 $[p_{(\text{Cluster } 2)} + 1]$ as the weighting term in the formula, it is possible to substitute the cumulative *t* values, *tt* $[tt_{(\text{Cluster } 2)} + t_{(\text{new triple})}]$. Consequently, under this method of weighting, the new formula for computation of the new row mean for Cluster 2 becomes:

$$[1/tt_{(\text{Cluster } 2)} + t_{(\text{new triple})}]$$

$$* [tt_{(\text{Cluster } 2)} * r_{(\text{Cluster } 2)} + t_{(\text{new triple})} * r_{(\text{new triple})}],$$

which is $[1/(45+15)] * [45 * 25 + 15 * 36]$, which equals 27.75, or 28 when rounded to the nearest integer.

The new column mean for Cluster 2 is computed as follows:

$$[1/tt_{(\text{Cluster } 2)} + t_{(\text{new triple})}]$$

$$* [tt_{(\text{Cluster } 2)} * c_{(\text{Cluster } 2)} + t_{(\text{new triple})} * c_{(\text{new triple})}],$$

which is $[1/(45+15)] * [45 * 40 + 15 * 25]$, which equals 36.25, or 36 when rounded to the nearest integer.

Thus, the mean for Cluster 2 under this method of weighting is now situated at row 28 and column 36.

The Choice of Weighting Method

Method 1, MacQueen's (1967) original algorithm, produces stable means as the number of assigned triples for each cluster increases. Method 2, weighting by cumulative fixation time, is the most stable of the two methods, and, as might be expected, produces discrete clusters and very stable cluster means as fixation time accumulates within each cluster. Which weighting method should be used? Confidence in the choice of the initial cluster means should be a deciding factor. If an experimenter is confident in the choice of the initial mean estimates—for instance, on the grounds that the peaks and troughs in the original CFT distribution are quite clear—

then the more stable Method 2 should be used. The CFT distribution depicted in Figures 3 and 4 is an example in which the peaks and troughs are fairly well distinguished, and one should have no hesitation about using Method 2 on such a distribution. On the other hand, if peaks in the original CFT distribution are not altogether clear, and the choice of initial means is difficult, then the use of Method 1 could provide a means of deciding between different sets of initial mean estimates. Because Method 1 is slightly less stable, the cluster means are more likely to depart from their original positions and to gravitate toward the highest peaks in the CFT distribution. The final mean positions produced by this method can then become the initial mean estimates for application of the more stable Method 2. As Hunt (1975) pointed out, a constructive algorithm such as the *k means* procedure can be justified only by results since it is not based on a criterion function. Accordingly, since results and comparisons of the effects of the two methods may be obtained by running the *k means* option in Program CFT _ Analyzer (Appendix B) on the dummy CFT distribution provided on floppy disk, users of the procedures are encouraged to experiment and compare. For example, in this very clear-cut distribution with four main peaks, it will be noticed that under both methods, even in the extreme case in which the initial mean locations are set at the four corners of the distribution and as far from the highest peaks as possible, *the means still migrate toward and terminate upon the highest peaks and centers of gravity of the distribution.*

As well as supplying a table of the locations of the final means of clusters, Program CFT _ Analyzer produces a topographical representation of the cluster solution. As each triple in a CFT distribution is assigned to a cluster, its *t* value is exchanged for the number of the cluster to which the triple has been assigned. Thus, using the CFT distribution presented in Figure 3 and the initial estimates of the cluster means circled in Figure 3 and set out in Table 1, Program CFT _ Analyzer would produce the results depicted in Table 2 and Figure 10 using Method 2 (CFT weighting). For refinements and recent developments of this clustering technique, the reader is referred to Hunt (1975, pp. 99-100).

Interpreting and Using the Cluster Solution

What extra information is added to or new light shed upon the CFT distribution by the cluster analysis? As pointed out above, the procedures for representing the CFT distribution in two and three dimensions, together with the decisions on coarseness and thresholding, all involve a fair amount of subjective judgment. The *k means* cluster analysis, like any cluster analysis, also involves subjective judgment, but, nonetheless, the assignment of triples to clusters has been executed by a random process and the computation of the final mean positions and membership of clusters carried out by a quantitative procedure. With the CFT distribution partitioned into clusters, an experimenter can indicate clearly the stimuli or regions of

Table 2
Final Cluster Means and Other Statistics for the *k means*
Cluster Analysis (Type 2, CFT Weighting) of the
CFT Distribution Presented in Figure 3

Cluster	Row	Column	Total <i>t</i>	<i>M t</i>	Triples
1	21	17	3516	8.81	399
2	14	37	2675	7.47	358
3	30	27	4290	15.43	278
4	34	36	3161	10.86	291
5	53	38	4052	8.64	469
6	54	19	3533	6.93	510

stimuli receiving most attention and, in consequence, proceed to mount arguments for and against the rejection of hypotheses about the relative discriminability of the stimuli or differentiating capacity of the stimulus regions. In the case of the pilot study example, there were no *precise* hypotheses about the location of clusters, but it is clear that the clusters are located on regions of the screen at which the stimulus lines could appear, and thus provide support for the contention that, under the particular tunnel-vision-producing conditions, CFT distributions are not random, but are determined by stimulus configurations. Given that experimentation following this pilot study used the same methods and analyses in an investigation of the relative differentiating capacity of features or regions within large alphabetic characters, this was a necessary and important result in itself. Finally, as already argued, the more precise hypotheses proposed in eye-movement investigations of cognitive processes, such as those cited above, could be explored and rigorously tested by the described clustering techniques.

CLASSIFICATION BY INFORMATION MEASURE

How to Avoid Initial Mean Estimates

The *k means* cluster analysis requires and depends upon stipulation of the number of clusters and their initial mean estimates. For many reasons, a theorist or experimenter may wish to escape this commitment, involving, as it does, subjective judgment and possible bias. Furthermore, in some cases, a CFT distribution may not reveal clear and unambiguous peaks and troughs, and the theorist may be quite unsure about initial mean estimates. In other cases, the CFT distribution may allow several sets of initial mean estimates, some of which may be favorable and some unfavorable to theories under test. One important classification procedure that eludes the requirement of initial mean estimates is that of Wallace and Boulton (1968a, 1968b). Wallace and Boulton defined what they called "the information measure" and suggested that a classification may be regarded as a method of representing more briefly the information contained in the set of *D* "measurements" of some set of *S* "things."

A Sample CFT Distribution for Classification

By way of illustration, consider the new CFT distribution presented in Figure 11. These data were recorded

Figure 10. A topographical representation of the six-mean solution provided by a k means cluster analysis (Type 2, weighting by CFT) of the CFT distribution presented in Figure 3. Each number indicates the cluster to which that particular triple was assigned, and the circled double stars mark the final cluster mean positions.

Figure 11. A cumulative CFT distribution across 12 subjects recognizing all 26 large uppercase alphabetic characters under tunnel-vision-producing conditions.

by 12 subjects who were asked to recognize very large uppercase alphabetic characters under the same tunnel-vision-inducing stimulus conditions already described for the line-detection experiment. Given various start points on the large bright screen and 5 sec to view each letter, the subjects were forced into a serial scan of the letter elements in order to recognize the stimulus. The resulting CFT distribution represents the cumulative fixation times of the 12 subjects recognizing the 26 uppercase alphabetic stimuli across a number of trials on each letter.

In this example, the set of S things under consideration is the set of 18,404 data points that occupy the 3,850 locations in the 79×52 (number border excluded) CFT distribution. The set of measurements (attribute values) of each thing comprises its row and column indices. For example, starting at the top left-hand corner of Figure 11, one encounters one thing having the attribute values row 2 and column 22, then another thing with attribute values row 2 and column 24, then two things with attribute values row 2 and column 27, and so on. At the highest peak in the center of the distribution, there are 82 things with the attribute values row 38 and column 27. Thus, the row and column indices become the attribute values of the things to be classified, and the t values become their frequency of occurrence; the total of the frequencies is 18,404.

The Best Classification Is the Shortest Message

Wallace and Boulton (1968a, 1968b) then considered the descriptions of the data points, or things, as a set of S messages which, when linked together, convey the total amount of information in the CFT distribution. Accordingly, if the distribution of t values in the matrix is uniform, one cannot encode the data points more briefly than by a simple list of the measured values in the manner described. That is, if there is no basis for grouping the S things into subclasses of things that are alike, or, in this particular context, no grounds for asserting that groups of fixations within the total CFT distribution form clusters, because of their density and proximity, then the total information in the CFT distribution cannot be expressed in a shorter form than by a complete list of the things and their attribute values. However, Wallace and Boulton pointed out that Shannon (1948) showed that, where messages may be regarded as each nominating the occurrence of a particular event among a universe of possible events, the information needed to record a series of such messages is minimized if the messages are encoded so that the length of each message is proportional to minus the logarithm of the relative frequency of the event that it nominates. The required message length is greatest when all the frequencies are equal. However, in the case of eye-movement data, where the expected CFT distribution is nonuniform, Shannon's theorem allows a reduction in the total message length by using a brief encoding for positions in those regions where CFT is greatest.

In very simple terms, Wallace and Boulton (1968a, 1968b) said that the 18,404 things in the CFT distribu-

tion (Figure 11) can be described by a string of 18,404 messages about their attributes, but if the distribution of things is nonuniform, then it may be possible to economize by including, in the total message, a message describing the distribution of things. The *best classification* of any set of S things will be the *shortest message* that contains all the information carried by the descriptions of the S things. It is such an economy, by appropriate classification of the things, that the Wallace and Boulton procedure seeks for any set of S things. Thus, the procedure encodes the $S \times D$ measurements in the CFT distribution into a message having five components: (1) the number of classes; (2) the dictionary of class names; (3) a description of the distribution function for each class; (4) for each thing, the name of the class to which it belongs; and (5) for each thing, its attribute values in the code set up for the class. In the classification of eye-movement data, the value of Wallace and Boulton's procedure lies in its production, *without intervention by the user of the procedure*, of the number of classes (clusters), as well as a description of the cluster distributions.

Program SNOB

Wallace and Boulton (1968a, 1968b) embodied their classification procedure in a program called SNOB, originally written in ALGOL and run on a KDF9 computer at the University of Sydney. For input, SNOB requires (1) the number of things (S), (2) the number of attributes, (3) a description of each thing in terms of its attributes, and (4) the nature of the attributes (discrete or continuous). In applying SNOB to the CFT distribution in Figure 11, a problem immediately arose in connection with the amount of information the program can handle; 18,404 things were beyond its specifications. Consequently, before presenting the CFT distribution to the program for analysis, it was further reduced by dividing each t value in Figure 11 by 4 to produce 5,272 things. It should be appreciated that this reduction resulted in further loss of information from the original distribution. (To conform to printout limitations, the original distribution already had been reduced.) Table 3 shows the class solution produced by program SNOB from the reduced CFT distribution depicted in Figure 12. Program SNOB determines the optimum number of classes, but it is not possible to recover the actual points or things allocated to each class, and, therefore, it is not possible to provide a topographical representation of the class solution as it

Table 3
The Wallace and Boulton Class Solution for the Reduced CFT Distribution Presented in Figure 12

Class	M Row	SD Rows	M Column	SD Columns	"Things"
Total	38.21	12.76	27.29	9.50	5272
1	38.03	12.06	14.24	3.02	1168
2	17.55	4.70	26.36	3.44	395
3	39.88	13.88	39.95	3.22	1389
4	58.75	5.38	27.78	3.70	384
5	37.25	5.83	27.33	3.28	1936

79 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 79

Figure 12. A reduced version of the CFT distribution presented in Figure 11. To produce this reduced version, each t value in the original CFT distribution has been divided by 4 and rounded. The circled t values are the means of the Wallace and Boulton five-class solution for this reduced CFT distribution and the mean of the total distribution.

is with *k means*. However, the means of the classes produced by SNOB and the mean of the distribution have been circled on Figure 12.

Boulton (personal communication) suggested that had it been possible to make use of the unreduced CFT distribution (Figure 11) instead of the reduced version (Figure 12), it is likely that more classes would have been created, but that computing time would have become rather excessive. By way of comparison, Figure 13 is the topographical representation of a *k means* (Method 2) analysis of the unreduced CFT distribution (Figure 11) where the nine initial estimates of the means were taken as nine isolated high peaks in the distribution (circled on Figure 11).

Interpreting and Using the Class Solution

Class solutions may be interpreted in a manner similar to the *k means* cluster solutions. The five-class solution for the reduced distribution indicates that subjects, in their attempts to recognize letter stimuli, directed most attention to elements or features of the letters placed at center, center right, center left, top center, and bottom center. This information was, in turn, used in inferences about the feature analysis of the letters used by the subjects and in tests of a theory (Sutcliffe, 1986) that predicts which features of letters are most differentiating and, in consequence, what weights should be assigned to features during recognition. Other possible applications of the information measure in the assessment of eye-movement data are similar to those already discussed in connection with the *k means* technique, and depend largely on the experimenter's purposes, together with the type and precision of the hypotheses under test.

Information Measure or *k means*?

Detailed arguments and reasons for using the variants of *k means* have already been made, and the benefits of the information measure over these procedures have been outlined. It should, however, be noted that the five-class solution (Figure 12) for the reduced CFT distribution is, in fact, a subset of the *k means* nine-cluster solution (Figure 13) for the unreduced CFT distribution, and that were SNOB able to handle this unreduced CFT distribution, the two solutions might converge even further. This is an interesting finding and may indicate a significant convergence in solutions produced by these seemingly quite different procedures. Nonetheless, there are aspects of both procedures, some academic and some practical, that need to be considered before a choice is made. Classification by information measure is a preferable technique on purely theoretical grounds. It removes the requirement of estimating the number of classes and class means and, in so doing, introduces a level of objectivity not found in other classification procedures. It is particularly useful in cases in which a theorist may have difficulty in specifying initial estimates. However, SNOB has difficulty handling large CFT distributions and, although the algorithm is described in detail in Wallace and Boulton (1968a), the program itself is not available generally. On

the other hand, programs for representing CFT distributions in two and three dimensions and applying *k means* to these are readily available (see below).

Conclusions

Fixation time on stimulus displays can be accumulated and displayed in two and three dimensions to reveal regions of high and low attention. One of the benefits of treating eye-movement data in this manner is that CFT distributions can be subjected to clustering techniques that provide quantitative means of partitioning the distributions. With CFT distributions partitioned into regions of high and low attention, an experimenter can set about testing hypotheses concerning which stimuli or regions of stimuli should receive most attention during inspection or recognition by subjects. Additionally, the methods reported here allow fixation time to be accumulated over many stimuli, trials, and subjects to reveal general areas of high fixation density and fixation time on stimulus displays, and, as such, not only invest eye-movement indices of cognitive processes with greater reliability, but also increase the range of quantitative methods available for the assessment of eye-movement data.

AVAILABILITY OF PROGRAMS

Program CFT_Analyzer and its associated module "EyemoveLib," with the input files of dummy data for demonstration ("Eyemove_Data" and "Kmeans_Data"), run successfully on Macintosh 512K Enhanced, Plus, SE, and Macintosh II (where three-dimensional images are produced in color), and are available on a 3.5-in. floppy disk.² True BASIC follows the proposed ANSI standard for advanced BASIC, and programs written for the Macintosh can easily be ported to other computers. Send requests for copies together with \$15 (U.S.) to the author. The program can be supplied in versions that suit particular needs. An uncompiled version that requires True BASIC Version 2.0 or later for the Macintosh, together with the True BASIC Programming Libraries MacTools and PictLib (standard with Version 2.0) and the add-on 3-D Graphics Toolkit. This version will allow the user to edit the code and customize the algorithms. For those who do not have True BASIC for the Macintosh, a "double clickable" version that runs from the desktop can be supplied. This version is compiled, and editing of the program is not possible. Please stipulate which version is required.

The algorithm embodied in Program SNOB is described in detail by Wallace and Boulton (1968a). Requests for information about the program and reprints of papers should be addressed to C. S. Wallace, Department of Computer Science, Monash University, Clayton, Victoria 3168, Australia.

PROGRAM DEVELOPMENT

New versions of program CFT_Analyzer and module "EyemoveLib" are currently under development. These

Figure 13. A topographical representation of the nine-mean solution provided by a *k* means cluster analysis (Type 2, weighting by CFT) of the CFT distribution presented in Figure 11. Each number indicates the cluster to which that particular triple was assigned, and the circled double stars mark the final cluster mean positions.

will include routines for plotting two-dimensional contour maps and smoothed three-dimensional surfaces of CFT distributions. Although the present versions employ some of the standard Macintosh user interface (dialog boxes, beeps, etc.), a complete version that contains all the options and runs from the desktop is also under development. A Macintosh version of Wallace and Boulton's (1968a) program for classification by information measure is also being developed.

REFERENCES

- BAKER, M. A., & LOEB, M. (1973). Implications of measurement of eye fixations for a psychophysics of form perception. *Perception & Psychophysics*, **13**, 185-192.
- BLACKITH, R. E., & REYMENT, R. A. (1971). *Multivariate morphometrics*. London: Academic Press.
- HUNT, E. B. (1975). *Artificial intelligence*. New York: Academic Press.
- JOHNSTON, E. G., & ROSENFELD, A. (1975). Digital detection of pits, peaks, ridges and ravines. *IEEE Transactions on Systems, Man & Cybernetics*, **5**, 472-480.
- JUST, M. A., & CARPENTER, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*, **8**, 441-480.
- LATIMER, C. R. (1978). Eye movement indices of feature extraction and feature weighting in letter recognition. *Dissertation Abstracts International*, **40**, 952B. (University Microfilms No. 79-18585)
- MACKWORTH, N. H., & MORANDI, A. J. (1967). The gaze selects informative details within pictures. *Perception & Psychophysics*, **2**, 547-552.
- MACQUEEN, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Statistics and Probability* (pp. 281-297). Berkeley, CA: University of California Press.
- NODINE, C. F., & STEUERLE, N. L. (1973). Development of cognitive strategies for differentiating graphemes. *Journal of Experimental Psychology*, **97**, 158-166.
- PASNAK, R. (1971). Pattern complexity and response to distinctive features. *American Journal of Psychology*, **84**, 235-245.
- SCINTO, L. F. M., & BARNETTE, B. D. (1986). An algorithm for determining clusters, pairs or singletons in eye-movement scan-path records. *Behavior Research Methods, Instruments, & Computers*, **18**, 41-44.
- SHANNON, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, **27**, 379-423, 623-656.
- SUTCLIFFE, J. P. (1986). Differential ordering of objects and attributes. *Psychometrika*, **51**, 209-240.
- TICHOMIROV, G. K., & POSNYANSKAYA, E. D. (1966). An investigation of visual search as a means of analyzing heuristics. *Voprosy Psikhologii*, **12**, 39-53.
- VURPILLOT, E. (1968). The development of scanning strategies and their relation to visual differentiation. *Journal of Experimental Child Psychology*, **6**, 632-650.
- WALLACE, C. S., & BOULTON, D. M. (1968a). *Classification by information measure* (Tech. Rep. No. 48). Sydney, Australia: University of Sydney, School of Physics, Basser Computing Department.
- WALLACE, C. S., & BOULTON, D. M. (1968b). An information measure for classification. *Computer Journal*, **11**, 185-194.
- WINIKOFF, A. (1967). *Eye movements as an aid to protocol analysis of problem solving behavior*. Unpublished doctoral dissertation, Carnegie-Mellon University, Pittsburg, PA.

NOTES

1. True BASIC is a trademark of True BASIC Inc., 39 S. Main Street, Hanover, NH 03755.
2. Macintosh, ImageWriter, LaserWriter, and MacWrite are trademarks of Apple Computer, Inc., P.O. Box 1143, Cupertino, CA 95014.
3. DeskPaint is a trademark of Zedcor Inc., 4500 E. Speedway, Suite 22, Tucson, AZ 85712-5305.
4. PostScript is a trademark of Adobe Systems, Inc., P.O. Box 7900, Mountain View, CA 94039-7900.
5. SendPS is a trademark of Cricket Software, 30 Valley Stream Parkway, Great Valley Corporate Center, Malvern, PA 19355.

APPENDIX A

Program CFT_Analyzer

```

!
!------(PROGRAM CFT_Analyzer)-----
!
!                               VERSION 1.0
!
! Written by Dr.C.R. Latimer, Department of Psychology,
! University of Sydney, NSW 2006, Australia. July 1988.
!
! Language: True BASIC, True BASIC Inc., 39 South Main Street,
! Hanover, N.H. 03755, USA.
!
! Program CFT_Analyzer sequences calls to subroutines and Libraries
! in Module "EyemoveLib" for 3_D plotting and k_means cluster analyses
! of cumulative-fixation-time (CFT) distributions.
!
!
! LIBRARY "EyemoveLib"
! DO
!     CALL MacSysBeep(.25)
!     PRINT
!     PRINT" -----ooo MAIN MENU ooo----- "
!     PRINT
!     PRINT"1. Cumulative Fixation Time (CFT) analysis.....type 1"

```

APPENDIX A (Continued)

```

PRINT"2. k_means cluster analysis.....type 2"
PRINT"3. Stop the program.....type 3"
INPUT Path
SELECT CASE Path
CASE 1          !CFT analysis
  CALL Cft_Controller
  DO
    CALL MacSysBeep(.25)
    PRINT
    PRINT" -----ooo CFT MENU ooo-----  "
    PRINT
    PRINT"1. Display the CFT and Reduced CFT distributions.  NOTE: On "
    PRINT" completion, the program will stop so that you can view "
    PRINT" the distributions.....type 1"
    PRINT"2. Print the CFT and reduced CFT distributions. Make sure "
    PRINT" that your printer is switched on.....type 2"
    PRINT"3. Save and view from various angles and distances the 3-D"
    PRINT" plot of the reduced CFT distribution.....type 3"
    PRINT"4 Copy to Clipboard and view from various angles and"
    PRINT" distances the 3-D Plot of the reduced distribution.....type 4"
    PRINT"5. Print and view from various angles and distances the 3-D"
    PRINT" plot of the reduced CFT distribution. Make sure that"
    PRINT" your printer is switched on.....type 5"
    PRINT"6. Return to Main Menu.....type 6"
    INPUT Choice
    SELECT CASE Choice
    CASE 1          ! Display cft distributions and statistics
      CALL Display_Cft(#0)
      EXIT DO
    CASE 2          ! Print cft distributions and statistics
      OPEN#5:Printer
      SET#5:MARGIN 120
      CALL Display_Cft(#5)
    CASE 3          ! Plot and save 3-D images of reduced distribution
      CALL Three_D_Plotter(Choice)
    CASE 4          ! Plot and Clipboard 3_D images
      CALL Three_D_Plotter(Choice)
    CASE 5
      CALL Three_D_Plotter(Choice) !Plot and print 3-D image
    CASE ELSE      ! Return to Main Menu
      EXIT DO
    END SELECT
    IF Choice = 1 THEN
      EXIT DO
    END IF
    CLOSE#5
  LOOP
CASE 2          ! k_means analysis
  CALL K_Means_Controller
  DO
    PRINT
    PRINT" -----ooo K_MEANS MENU ooo-----  "
    PRINT
    PRINT"1. Display the k_means solution on screen .....type 1"
    PRINT"2. Print the k_means solution. Make sure that your printer "
    PRINT" is switched on.....type 2"
    PRINT"3. Exit to Main Menu.....type 3"

```

APPENDIX A (Continued)

```

INPUT Choice
SELECT CASE Choice
CASE 1
    CALL Display_Clusters(#0)
    EXIT DO
CASE 2
    OPEN#5:Printer
    SET#5:MARGIN 120
    CALL Display_Clusters(#5)
    CLOSE#5
CASE ELSE
    EXIT DO
END SELECT
LOOP
CASE ELSE
    EXIT DO
END SELECT
CALL Closure
IF Choice=1 THEN
    EXIT DO
END IF
LOOP
STOP
END

!------(MODULE EyemoveLib)
!
! Version 1.0
!
! Written by Dr. C.R. Latimer, Department of Psychology
! University of Sydney, NSW 2006, Australia. July 1988.
!
! Language: True BASIC, True BASIC Inc., 39 South Main Street,
! Hanover, N.H. 03755, USA.
!
! The module contains routines for two- and three-dimensional
! plotting of cumulative-fixation-time (CFT) data and k_means
! cluster analysis. The uncompiled version of this module
! requires that the True BASIC Library MacTools* be loaded
! prior to use and that Libraries my3dlib* PictLib* be available.
!
! Main Variables
!
! Cft(,)          array to hold the CFT distribution
! Reduced_Cft(,)  array to hold the reduced CFT
! Zed_Values(,)   array to hold transposed z coordinates
! Clusters$(,)    array to hold k-means cluster solution
! Distances(,)    array to hold mean distances for k_means
! Initials(,)     array to hold initial mean coordinates
! Rows, Cols      Row and Column dimensions of Cft(,)
! Red_Rows,Red_Cols Row and Column dimensions of Reduced_Cft(,)
! Z_Size          Size of Z_Values array
! Threshold       Flag and degree of CFT thresholding
! X_Max           Maximum X_coordinate in CFT distribution
! Y_Max           Maximum Y_coordinate in CFT distribution
!

```

APPENDIX A (Continued)

MODULE EyemoveLib

```

!
LIBRARY "my3dlib*", "Pictlib*"
SHARE Cft(50,50),Reduced_Cft(20,20),Zed_Values(50)
SHARE Clusters$(50,50),Distances(10,2),Initials(10,5)
SHARE X1,Y1,X_Max,Y_Max,Rows,Cols,Red_Rows,Red_Cols,Z_Size,Threshold
SHARE Maximum,Minimum,Mean_Cft,Cft_Std,Q_Rows,Q_Cols,Rem_Rows,Rem_Cols
SHARE X_Start,X_Term,Y_Start,Y_Term,X_Step,Y_Step,Z_Step
SHARE ClusterType,k,Row,Col,Zeros,Intran,FileName$,Restart$
SHARE #1,#2,#3,#6,#7,#8,#10
!
!------(Subroutine CFT_Controller)-----
!
! This subroutine handles all housekeeping and sequences calls to
! other routines during plotting and analysis of CFT distributions
!
SUB Cft_Controller
DO
CALL MacSysBeep(.25)
INPUT prompt"Type the name of your data file ":FileName$
WHEN ERROR IN      ! Error trap for bad file name or hidden file
OPEN #1:name FileName$, access input, organization text, create old
EXIT DO
USE
CALL MacSysBeep(.25)
CALL MacSysBeep(.25)
PRINT"File not accessible !...Is it on the desktop ?"
INPUT PROMPT"Try again...type y; Stop the program...type n": What$
IF What$="n" THEN
STOP
END IF
END WHEN
LOOP
CALL MacSysBeep(.25)
INPUT PROMPT "Type Maximum X- then Maximum Y-coordinates (Integers)":X_Max,Y_Max
LET Rows=2*Y_Max+1
LET Cols=2*X_Max+1
MAT REDIM Cft(Rows, Cols) ! Redimensioning to user requirements
DO ! Error trap for faulty dimensions
CALL MacSysBeep(.25)
INPUT PROMPT "ROWS then COLS in reduced distribution ?": Red_Rows, Red_Cols
IF Red_Cols <= Cols AND Red_Rows <= Rows THEN
EXIT DO
ELSE
CALL MacSysBeep(.25)
CALL MacSysBeep(.25)
PRINT"Reduced dimensions exceed original";("";Rows;Cols;") !"
INPUT PROMPT"Try again...type y; Stop the program...type n":Which$
IF Which$="n" THEN
STOP
END IF
END IF
LOOP
MAT REDIM Reduced_Cft(Red_Rows,Red_Cols)
LET Z_Size=Red_Rows*Red_Cols
MAT REDIM Zed_Values(Z_Size)

```

APPENDIX A (Continued)

```

CALL Xys_To_Rowscols    !Reading in Datafile
CALL MacSysBeep(.25)
INPUT PROMPT "What threshold is required for the Cft distribution ?":Threshold
IF Threshold > 0 THEN
    CALL Thresholder
END IF
CALL Reduce
CALL MacSysBeep(.25)
INPUT PROMPT "Save the Cft distribution as a file ?, type y or n":CFT$
IF CFT$="y" THEN
    CALL StoreFile(#2,Cft(,),Rows,Cols)
END IF
CALL MacSysBeep(.25)
INPUT PROMPT "Save the reduced distribution as a file ?, type y or n":REDCFT$
IF REDCFT$="y" THEN
    CALL StoreFile(#3,Reduced_Cft(,),Red_Rows,Red_Cols)
END IF
IF Threshold = 0 THEN
    CALL Summary_Stats
END IF
END SUB
!
!------(Subroutine K_Means_Controller)-----
!
! This subroutine handles housekeeping and sequences calls for k-means
! analyses of CFT distributions
!
SUB K_Means_Controller
    LET Restart$="try_again"
    DO WHILE Restart$="try_again"
        DO
            CALL MacSysBeep(.25)
            INPUT prompt "Type the name of your CFT distribution file ":FileName$
            WHEN ERROR IN      ! Error trap for bad file name or hidden file
                OPEN#8:name FileName$, organization text, access input, create old
            EXIT DO
        USE
            CALL MacSysBeep(.25)
            PRINT "File not accessible !...Is it on the desktop ?"
            INPUT PROMPT "Try again...type y; Stop the program...type n": What$
            IF What$="n" THEN
                STOP
            END IF
        END WHEN
    LOOP
    CALL MacSysBeep(.25)
    INPUT prompt "Number of ROWS then Columns of your CFT matrix ":Rows,Cols
    CALL MacSysBeep(.25)
    INPUT prompt "Type the number of initial mean estimates ":k
    MAT redim Cft(Rows,Cols)
    MAT redim Clusters$(Rows,Cols)
    MAT redim Distances(k,2)
    MAT redim Initials(k,5)
    FOR i=1 to k
        DO
            CALL MacSysBeep(.25)
            PRINT "For Mean..."i;"Type the ROW then COL coordinate then CFT value"

```

APPENDIX A (Continued)

```

        INPUT Initials(i,1),Initials(i,2),Initials(i,3)
        IF Initials(i,1)>Rows OR Initials(i,2)>Cols THEN
            CALL MacSysBeep(.25)
            PRINT"Mean coordinates exceed matrix bounds...Try again"
        ELSE
            EXIT DO
        END IF
    LOOP
    LET Initials (i,4)=0
    LET Initials(i,5)=Initials (i,3)
NEXT i
CALL MacSysBeep(.25)
PRINT"Which Clustering algorithm would you like ?"
PRINT"1. Original MacQueen .....type 1"
PRINT"2. CFT-Weighted.....type 2"
INPUT ClusterType
CALL K_Means
LOOP
CALL MacSysBeep(.25)
INPUT prompt"Save the k-means solution as a file ? Type y or n ":KM$
IF KM$="y" THEN
    LET Name$=MacPutFile$(0,0,"Save k_means as", "", "")
    IF Name$ <> "" THEN
        OPEN #10:name Name$, access outin, organization text, create newold
        ERASE#10
        FOR i=1 TO Rows
            FOR j=1 TO Cols
                PRINT#10:Clusters$(i,j)
            NEXT j
        NEXT i
    END IF
END IF
END SUB
!
!------(Subroutine StoreFile)-----

!This subroutine stores files when required by the user
!
SUB StoreFile(#6,Array(),r,c)
    LET Name$=MacPutFile$(0,0,"Save distribution as", "", "")
    IF Name$="" THEN EXIT SUB
    OPEN #6:name Name$, access outin, organization text, create newold
    ERASE#6
    FOR i=1 TO r
        FOR j=1 TO c
            PRINT#6:Array(i,j)
        NEXT j
    NEXT i
END SUB
!
!------(Subroutine Thresholder)-----

! This subroutine deletes entries in the CFT distribution below the value
! of "Threshold" input by the user. In this way, higher CFT values are
! enhanced during plotting.
!
```

APPENDIX A (Continued)

```

SUB Thresholder
  FOR i=1 TO Rows
    FOR j=1 TO Cols
      IF Cft(i,j) < Threshold THEN LET Cft(i,j)=0
    NEXT j
  NEXT i
END SUB
!
!------(Subroutine Xys_To_Rowscols)-----
!
! This subroutine converts raw x-y coordinates to row and column
! coordinates for matrix Cft(.)
!
SUB Xys_To_Rowscols
  DO
    WHEN ERROR IN      ! Error trap for too large x & y maxima
      MAT Cft=ZER
      LET X_Max = X_Max+1
      LET Y_Max = -1*(Y_Max+1)
      DO WHILE MORE #1
        INPUT#1:X
        INPUT#1:Y
        INPUT#1:Tim
        LET Col= X+X_Max
        LET Row = -1*(Y+Y_Max)
        LET Cft(Row,Col)=Cft(Row,Col)+Tim
      LOOP
      EXIT DO
    USE
      CALL MacSysBeep(.25)
      PRINT"Your x and y maxima must be too large !"
      PRINT"Try again....type y; Stop the program....type n"
      INPUT What$
      SELECT CASE What$
        CASE "y"
          CALL MacSysBeep(.25)
          INPUT PROMPT"Type Max X- then Max Y-coordinates(Integers)":X_Max, Y_Max
          LET Rows=2*Y_Max+1
          LET Cols=2*X_Max+1
          MAT REDIM Cft(Rows, Cols) ! Redimensioning to user requirements
          RESET#1:begin
        CASE ELSE
          STOP
      END SELECT
    END WHEN
  LOOP
  FOR i=1 TO Rows
    FOR j=1 TO Cols
      LET Cft(i,j)=TRUNCATE(Cft(i,j),0)
    NEXT j
  NEXT i
END SUB
!
!------(Subroutine Summary_Stats)-----
!
! This subroutine computes some simple descriptive statistics of the
! CFT distribution

```


APPENDIX A (Continued)

```

!
SUB Summary_Stats
  LET Minimum=Cft(1,1)
  LET Maximum=Cft(1,1)
  LET Sum=0
  LET n=Rows*Cols
  LET Sums_OF_Squares=0
  FOR i=1 TO Rows
    FOR j=1 TO Cols
      IF Cft(i,j) < Minimum THEN LET Minimum=Cft(i,j)
      IF Cft(i,j) > Maximum THEN LET Maximum=Cft(i,j)
      LET Sum=Sum+Cft(i,j)
      LET Sums_of_Squares=Sums_of_Squares+Cft(i,j)^2
    NEXT j
  NEXT i
  LET Mean_Cft=Sum/n      ! Mean
  LET Sums_Sq_Dev=Sums_of_Squares-Sum^2/n
  LET Cft_Std=SQR(Sums_Sq_Dev/(n-1)) !Standard deviation
END SUB

!
!------(Subroutine Reduce)-----
!
! This subroutine reduces the original Cft array to the Reduced_Cft
! array for 3_D plotting. Like the original Cft dimensions, the
! reduced dimensions are determined by the user.
!
SUB Reduce
  MAT Reduced_Cft=Zer
  CALL DIVIDE(Rows,Red_Rows,Q_Rows,Rem_Rows)
  CALL DIVIDE(Cols,Red_Cols,Q_Cols,Rem_Cols)
  LET kk=1
  LET LL=Q_Rows
  FOR i=1 TO Red_Rows
    LET mm=1
    LET nn=Q_Cols
    FOR j=1 TO Red_Cols
      FOR n=kk TO LL
        FOR k=mm TO nn
          LET Reduced_Cft(i,j)=Reduced_Cft(i,j) + Cft(n,k)
        NEXT k
      NEXT n
      IF j<Red_Cols-1 THEN
        LET mm=mm+Q_Cols
        LET nn=nn+Q_Cols
      ELSE
        LET mm=mm+Q_Cols
        LET nn=nn+Q_Cols+Rem_Cols
      END IF
    NEXT j
    IF i<Red_Rows-1 THEN
      LET kk=kk+Q_Rows
      LET LL=LL+Q_Rows
    ELSE
      LET kk=kk+Q_Rows
      LET LL=LL+Q_Rows+Rem_Rows
    END IF
  NEXT i
END SUB

```

APPENDIX A (Continued)

```

!
!------(Subroutine Three_D_Plotter)-----
!
! This subroutine handles all 3_D plotting of the CFT distribution
!
SUB Three_D_Plotter(Choice)
  LET X1=40          ! x coordinate of view
  LET Y1=-60         ! y coordinate of view
  LET Z1=160         ! z coordinate of view
  LET d=20           ! viewing distance
  LET Z_Divisor=35   ! column height control
  CALL Three_D_Loopvalues
  DO
    DO
      CALL Three_D_Changeview
      CALL PersWindow(-5,5,-5,5,0,10,S$)
      WHEN ERROR IN   ! Error trap for illegal coordinates
        CALL SetCamera3(X1,Y1,Z1,S$)
        CALL SetDistance3(d,S$)
      EXIT DO
    USE
      CALL MacSysBeep(.25)
      CALL MacSysBeep(.25)
      PRINT "Illegal coordinates !"
      PRINT "Try some more !"
      PRINT "X= ";X1;" Y= ";Y1;" Z= ";Z1;" Distance = ";d;" Z_Divisor = ";Z_Divisor
      INPUT PROMPT "Next X,Y,Z, distance and Z_Divisor ":X1,Y1,Z1,d,Z_Divisor
    END WHEN
  LOOP
  SELECT CASE Choice
  CASE 3
    CALL MacSysBeep(.25)
    INPUT PROMPT "Save the 3D plot as a file ?, type y or n":ANSWER$
    IF Answer$="y" THEN
      LET Name$=MacPutFile$(0,0,"Save the 3-D Plot as","", "")
      IF Name$<>"" THEN
        CALL Copy_Pictfile(name$)
      END IF
    END IF
  CASE 4
    CALL Copy_clipboard
  CASE 5
    CALL Copy_printer (0)
  END SELECT
  CLEAR
  LET Z_Step=0
  FOR Y = Y_Start TO Y_Term Step Y_Step
    FOR X = X_Start TO X_Term Step X_Step
      LET Z_Step=Z_Step+1
      LET Z=Zed_Values(Z_Step)/Z_Divisor
      CALL Block3(X-1.7,X-.3,Y-1.7,Y-.3,0,Z,"Yellow","Red","White",1,s$)
    NEXT X
  NEXT Y
  CALL Copy_done
  CALL MacSysBeep(.25)
  LINE INPUT PROMPT "Press Return to continue.":s$
  CALL MacSysBeep(.25)

```

APPENDIX A (Continued)

```

INPUT PROMPT "Change view ? Type y or n ?":V$
SELECT CASE V$
CASE "y"
    PRINT "X= ";X1;" Y= ";Y1;" Z= ";Z1;" Distance = ";d;" Z_Divisor = ";Z_Divisor
    CALL MacSysBeep(.25)
    INPUT PROMPT "Next X,Y,Z, distance and Z_Divisor":X1,Y1,Z1,d,Z_Divisor
CASE ELSE
    EXIT DO
END SELECT
LOOP
END SUB
!
!------(Subroutine Display_Cft)-----
!
! This subroutine displays the Cft, Reduced_Cft distributions and
! descriptive statistics on the screen or printer.
!
SUB Display_Cft(#7)
    PRINT#7:
    PRINT#7:"Data File Analyzed: ",FileName$; "(";Rows;" , ";Cols;")"
    PRINT#7:
    MAT PRINT#7,USING"###":Cft
    PRINT#7:
    MAT PRINT#7,USING"###":Reduced_Cft
    IF Threshold = 0 THEN
        PRINT#7:
        PRINT#7: "Maximum Cft   ",Maximum
        PRINT#7: "Minimum Cft   ",Minimum
        PRINT#7: "Mean Cft     ",Mean_Cft
        PRINT#7: "Standard Deviation of Cfts ", Cft_Std
    END IF
END SUB
!
!------(Subroutine Display_Clusters)-----
!
! This subroutine displays the k_means solution and descriptive
! statistics on the screen or printer
!
SUB Display_Clusters(#7)
    SELECT CASE Clustertype
    CASE 1
        LET Algor$="MacQueen"
    CASE ELSE
        LET Algor$="CFT-Weighted"
    END SELECT
    PRINT#7:
    PRINT#7:"Data File Analyzed: ",FileName$; "(";Rows;" , ";Cols;")"
    PRINT#7:"Cluster Algorithm used: ",Algor$
    PRINT#7:
    PRINT#7:"Row Co-ord", "Col Co-ord", "Total-T", "Triples", "Mean T"
    MAT PRINT#7: Initials
    PRINT#7:
    PRINT#7:"Total Empty Spaces   ",Zeros
    PRINT#7:"Total Intransigent Triples ",Intran
    PRINT#7:
    MAT PRINT#7, USING"###":Clusters$
END SUB

```

APPENDIX A (Continued)

```

!
!------(Subroutine Three_D_Loopvalues)-----
!
! This subroutine computes the values and signs of loop values for
! plotting 3_D images
!
SUB Three_D_Loopvalues
  CALL DIVIDE(Red_Rows,2,Q_Rows,Rem_Rows)
  LET Y_Start=Q_Rows*2
  IF Rem_Rows=0 THEN
    LET Y_Term=Q_Rows*2-2
  ELSE
    LET Y_Term=Q_Rows*2
  END IF
  CALL DIVIDE(Red_Cols,2,Q_Cols,Rem_Cols)
  LET X_Start=Q_Cols*2
  IF Rem_Cols=0 THEN
    LET X_Term=Q_Cols*2-2
  ELSE
    LET X_Term=Q_Cols*2
  END IF
END SUB

!
!------(Subroutine Changeview)-----
!
! This subroutine, depending on the required view of the 3_D image,
! transposes the Reduced_Cft z coordinates for plotting
!
SUB Three_D_Changeview
  LET Z_Step=0
  IF X1>=0 AND Y1<=0 THEN ! View from bottom-right
    FOR i=1 TO Red_Rows
      FOR j=1 TO Red_Cols
        LET Z_Step=Z_Step+1
        LET Zed_Values(Z_Step)=Reduced_Cft(i,j)
      NEXT j
    NEXT i
    LET Y_Start=ABS(Y_Start)
    LET Y_Term=-1*ABS(Y_Term)
    LET X_Start=-1*ABS(X_Start)
    LET X_Term=ABS(X_Term)
    LET Y_Step=-2
    LET X_Step=2
  ELSEIF X1>0 AND Y1>=0 THEN !View from top-right
    FOR i=Red_Rows TO 1 Step -1
      FOR j=1 TO Red_Cols
        LET Z_Step=Z_Step+1
        LET Zed_Values(Z_Step)=Reduced_Cft(i,j)
      NEXT j
    NEXT i
    LET Y_Start=-1*ABS(Y_Start)
    LET Y_Term=ABS(Y_Term)
    LET X_Start=-1*ABS(X_Start)
    LET X_Term=ABS(X_Term)
    LET Y_Step=2
    LET X_Step=2
  END IF
END SUB

```

APPENDIX A (Continued)

```

ELSEIF X1<=0 AND Y1>0 THEN ! View form top-left
  FOR i=Red_Rows TO 1 Step -1
    FOR j=Red_Cols TO 1 Step -1
      LET Z_Step=Z_Step+1
      LET Zed_Values(Z_Step)=Reduced_Cft(i,j)
    NEXT j
  NEXT i
  LET Y_Start=-1*ABS(Y_Start)
  LET Y_Term=ABS(Y_Term)
  LET X_Start=ABS(X_Start)
  LET X_Term=-1*ABS(X_Term)
  LET Y_Step=2
  LET X_Step=-2
ELSE
  FOR i=1 TO Red_Rows !View from bottom-left
    FOR j=Red_Cols TO 1 Step -1
      LET Z_Step=Z_Step+1
      LET Zed_Values(Z_Step)=Reduced_Cft(i,j)
    NEXT j
  NEXT i
  LET Y_Start=ABS(Y_Start)
  LET Y_Term=-1*ABS(Y_Term)
  LET X_Start=ABS(X_Start)
  LET X_Term=-1*ABS(X_Term)
  LET Y_Step=-2
  LET X_Step=-2
END IF
END SUB
!
!------(Subroutine RandSelect)-----
!
! This subroutine handles the randomization of the sequence of triples
! for k_means cluster analysis
!
SUB RandSelect
  LET Row=(Rows-2)*RND+2
  LET Col=(Cols-2)*RND+2
  LET Row=INT(Row)
  LET Col=INT(Col)
END SUB
!
!------(Subroutine K_Means)-----
!
! This subroutine carries out the k_means cluster analysis of the CFT
! distribution
!
SUB K_Means
  RANDOMIZE
  DO
    WHEN ERROR IN ! Error trap for input problems
      RESET#8:Begin
      FOR i=1 TO Rows
        FOR j=1 TO Cols
          INPUT#8:Cft(i,j)
        NEXT j
      NEXT i
      LET Restart$="ok" ! Flagging no restart
    EXIT DO
  
```

APPENDIX A (Continued)

```

USE
  CALL MacSysBeep(.25)
  CALL MacSysBeep(.25)
  PRINT"Either File ";FileName$;" does not contain enough data"
  PRINT"or the dimensions of the CFT distribution are wrong"
  PRINT"or both!"
  INPUT PROMPT"Try again...type y; Stop the program...type n":Answer$
  IF Answer$="y" THEN
    LET Restart$="try_again"
    CALL Closure
    EXIT SUB
  ELSE
    STOP
  END IF
END WHEN
LOOP
LET Triples=(Rows-2)*(Cols-2)-k    !number of triples in distribution
MAT Clusters$=NUL$(Rows,Cols)
LET Zeros=0
LET Intran=0
FOR L=1 TO k
  LET i=Initials(L,1)
  LET j=Initials(L,2)
  LET Clusters$(i,j)=STR$(L)
  LET Cft(i,j)=99
NEXT L
LET n=1
DO until n = Triples
  CALL RandSelect
  SELECT CASE Cft(Row,Col)
  CASE 0
    LET Cft(Row,Col)=99
    LET Zeros=Zeros+1
    LET n=n+1
  CASE 99
    CALL RandSelect
  CASE ELSE
    LET n=n+1
    FOR L=1 TO k    !computing mean distances from triple
      LET Irow=Initials(L,1)
      LET Jcol=Initials(L,2)
      LET Distances(L,1)=SQR((Irow-Row)^2+(Jcol-Col)^2)    ! Pythagoras
      LET Distances(L,2)=0
    NEXT L
    LET Min=Distances(1,1)
    FOR L=2 TO k
      IF Distances(L,1)<=Min THEN LET Min=Distances(L,1)
    NEXT L
    LET Flag=0
    FOR L=1 TO k
      IF Distances(L,1)=Min THEN LET Distances(L,2)=100
      IF Distances(L,2)=100 THEN LET Flag=Flag+1
    NEXT L
    IF Flag>1 THEN LET Intran =Intran+1
  LET L=1

```

APPENDIX A (Continued)

```

DO until L> k
  SELECT CASE Distances(L,2)
  CASE 0
    LET L=L+1
  CASE ELSE
    LET Clusters$(Row,Col)=STR$(L)
    IF ClusterType =1 THEN !MacQueen clustering
      LET Temp1=Initials(L,4)*Initials(L,1)+Row
      LET Temp2=Initials(L,4)+1
      LET Temprow=ROUND(Temp1/Temp2)
      LET Temp1=Initials(L,4)*Initials(L,2)+Col
      LET Tempcol=ROUND(Temp1/Temp2)
    ELSE !CFT-weighted clustering
      LET Temp1=Initials(L,3)*Initials(L,1)+Row*Cft(Row,Col)
      LET Temp2=Initials(L,3)+Cft(Row,Col)
      LET Temprow=ROUND(Temp1/Temp2)
      LET Temp1=Initials(L,3)*Initials(L,2)+Col*Cft(Row,Col)
      LET Tempcol=ROUND(Temp1/Temp2)
    END IF
    LET Initials(L,1)=Temprow
    LET Initials(L,2)=Tempcol
    LET Initials(L,3)=Initials(L,3)+Cft(Row,Col)
    LET Initials(L,4)=Initials(L,4)+1
    LET Initials(L,5)=Initials(L,3)/Initials(L,4)
    LET Clusters$(Row,Col)=STR$(L)
    LET Cft(Row,Col)=99
  EXIT DO
END SELECT
LOOP
END SELECT
LOOP
FOR L=1 TO k
  LET Row=Initials(L,1)
  LET Col=Initials(L,2)
  LET Clusters$(Row,Col)="*"
NEXT L
FOR i=1 TO Rows ! Putting border around cluster solution
  FOR j=1 TO Cols
    IF i=1 OR i=Rows THEN LET Clusters$(i,j)=STR$(j)
    IF j=1 OR j=Cols THEN LET Clusters$(i,j)=STR$(i)
  NEXT j
NEXT i
END SUB
!
!-----Subroutine Closure-----
!
!This subroutine closes all files in preparation for return to Main Menu
!
SUB Closure
  CLOSE#1
  CLOSE#2
  CLOSE#3
  CLOSE#6
  CLOSE#7
  CLOSE#8
  CLOSE#10
END SUB
END MODULE

```

APPENDIX B

Documentation for Program CFT_Analyzer

Some General Remarks on the Program

1. When working with True BASIC, it is simplest to have on the desktop (not in folders) all programs, libraries, and data files to be used during a run.

2. After opening and before running the uncompiled version of CFT_Analyzer, the Library MacTools* must be loaded using the LOAD command, and the Libraries 3dlib* and PictLib* must be on the desktop.

DOCUMENTATION

Generally, Program CFT_Analyzer takes, as input, a text file of fixation x and y coordinates and fixation times and produces, as output, two-dimensional tables and three-dimensional plots of CFT as well as topographical representations of k means cluster solutions. The format of the text file for input is as follows: First fixation x coordinate then Return, First fixation y coordinate then Return, First fixation time then Return, Second fixation x coordinate then Return, Second fixation y coordinate then Return, Second fixation time then Return, and so on. The file "Eyemove_Data" on the demonstration disk is an example of the required format.

On opening, the user is presented with the Main Menu, which has three choices: (1) CFT analysis, (2) k means cluster analysis, (3) stop the program. After the user chooses (1), the program prompts for the name of the data file to be analyzed. For the demonstration, type "Eyemove_Data" then Return. Next, the program prompts for the maximum x and maximum y coordinates (integers) in the file. The program works for square and rectangular matrices, but assumes that both x - and y -axes are symmetrical about zero. For the demonstration, type "9,14" then Return.

Next, the program prompts for the row then column dimensions of the matrix to hold the reduced CFT distribution. If no reduction is required, simply input the dimensions of the original CFT distribution. For example, the demonstration distribution with maximum x and y of 9 and 14, respectively, will require unreduced matrix dimensions of 19 and 29. Thus, for an unreduced view of the demonstration distribution, type "29,19" then Return. It should be remembered that three-dimensional images of large unreduced matrices take a long time to display and print and must be viewed at a long distance before they will fit entirely on the Macintosh screen.

For a reduced view, it is best to choose reduced matrix dimensions which divide the unreduced matrix dimensions without a large remainder. After division, remaining t values in the unreduced matrix are simply added into cells in the last row and last column of the reduced matrix. For example, a reduced matrix of seven rows and six columns is appropriate for the demonstration 29×19 unreduced matrix, since this reduction leaves only one row and one column of the unreduced matrix as remainders. However, a reduced matrix of five rows and four columns would leave remainders of four and three, respectively. On a first run, the program is set to give a good view of a reduced matrix of seven rows and six columns, so for the demonstration, type "7,6" then Return.

The program then prompts for the use of the thresholding procedure on the CFT distribution. If thresholding is required, type the t value below which t values are to be removed from the CFT distribution. For no thresholding and for the demon-

stration, type "0" then Return. At this point, the program reads the input file and computes the CFT distribution with or without thresholding.

Next, the program asks if the CFT distribution is to be saved as a file. This is important since it is this distribution that is used as input for the k means procedures in the program. So, if k means analysis is required, this file must be saved. If yes, then type "y," and the standard Macintosh file-saving box appears and allows the name of the file to be typed. Next, the user is asked if the reduced distribution is to be saved as a file, and the same procedure is required. Following this, the CFT menu is displayed and the user can select from a number of options.

1. Display the CFT and reduced CFT distributions with statistics on the screen. This option is not available in the compiled version of the program since it requires the output window that goes along with True BASIC. Note also that this option will STOP the program so that the distributions can be viewed in the scrollable output window. True BASIC only remembers the last 50 lines of output, so, for large matrices, use Option 2.

2. Print the CFT and reduced CFT distributions with statistics.

3. View and save the three-dimensional image of the reduced (or unreduced) CFT distribution. Before plotting this image, the program will ask if the result is to be saved as a file: type "y" or "n." Typing "y" will bring up the usual Macintosh save box, and saved files can be opened in any Macintosh graphics applications that accept PICT format. After the three-dimensional image has been displayed, the user is asked to press Return to continue and then to type "y" or "n" for a view of the distribution from a different angle, distance, and height. The "n" response brings up the CFT menu. At the "y" response, the current viewing parameters are displayed, and the user is invited to type in new parameters: x , y , z , viewing distance, and $Z_Divisor$. After typing these and pressing Return, the opportunity to save the file is again given.

4. Copy the three-dimensional image to the clipboard. This option operates exactly the same way as Option 3 except that the image, instead of being saved in a file, is placed on the clipboard for pasting directly into graphics applications.

5. Print the three-dimensional image. Again, this option operates like Option 3 except that the image is sent directly to a printer. This option allows for the full exploitation of the printer's resolution and not only the screen resolution. For example, images sent to a LaserWriter (see Note 1) using this option, will print with the full LaserWriter resolution.

6. Return to the Main Menu.

Menu item 2 in the Main Menu will initiate the procedures for the k means cluster analysis. First, the user is asked to type the name of the input file. This again must be a standard Macintosh text file with each CFT value on a separate line. For the demonstration, type "Kmeans_Data." Next, the program asks for the row and column dimensions of the CFT distribution to be analyzed. For the demonstration, type "29,19" then Return. The program then asks for the number of initial mean estimates; for the demonstration, type "4" then Return. The program will then require the row, column, and CFT value for each mean estimate in turn. For the demonstration, type "9,6,58" Return; "9,15,81" Return; "22,6,69" Return; "22,15,84" Return. The user is then requested to choose the clustering algorithm (Original MacQueen or CFT-weighted) and is then given the opportunity to save the topographical representation of the solution as a text file. Thereafter, the k means menu is presented, and it is possible to either display the topographical representation on the screen (not available in the compiled version), print the solu-

tion directly, or return to the Main Menu. In all cases, the cluster statistics are given and the final mean positions are marked by double stars.

HINTS FOR RUNNING THE PROGRAM

The program has many error traps and will provide opportunities to correct matrix dimensions that are out of bounds, viewing angles that are illegal, etc. The upper bounds for dimensions of input matrices depend on the memory available in each computer. However, it should be remembered that the display and printing of very large CFT distributions can be quite lengthy.

Viewing the Three-Dimensional Images

Option 3 in the CFT menu, which allows viewing the three-dimensional images of a distribution, is quite easy to control and the program is set for a good view of the demonstration distribution. The best views are those that are elevated and from (in compass terms) NE, SE, SW, and NW. If an image does not fit completely onto the screen, the distance should be increased until it does. If the columns extend beyond the top of the screen or wrap around the bottom, the $Z_Divisor$ should be increased, since this reduces the CFT values in the distribution through uniform division by the input amount. If the columns extend off the screen and wrap around, it is best to let the display of the image continue even though this may take a while. The $Z_Divisor$ value can then be increased when display of the oversized image is completed. Stopping the program merely reinstates the inadequate $Z_Divisor$. The relationship between the parameters "Distance" and elevation (z) is tricky, and occasionally some combinations of values of these parameters will lead to an illegal view such as that which occurs when the view plane is behind the camera position. The program traps these errors, and new values may be input.

Images saved under Option 3 can be quite large, and some applications are unable to open them. However, while expensive graphics applications crash, the little Macintosh desk accessory Deskpaint is highly recommended and seems to be able to handle just about every image regardless of size.³

Testing the k means Procedures

One way to test the validity of the k means procedures is mentioned in the text. Input the name of the demonstration file provided on disk, "Kmeans_Data," and specify a Row dimension of "29" and a Column dimension of "19." Next, specify four mean estimates, with the means set at the four corners of the distribution: mean 1, "1,1,0"; mean 2, "1,19,0"; mean 3, "29,1,0"; mean 4, "29,19,0." Under both clustering algorithms, because the peaks and troughs in the demonstration distribution are well defined, the means gravitate toward and settle on the highest peaks.

Printing

Printers are notoriously temperamental machines, but printing the CFT distribution, the k means solution, and three-dimensional images on ImageWriter printers poses no problems (see Note 1). However, although printing the CFT distribution and the k means solution on a LaserWriter is also straightforward, the printing of the three-dimensional images can be a very lengthy business and prone to error generation. Because the three-dimensional images are fairly complex, they take a long time to send to the printer and, consequently, can generate

"Offending Command, Error Timeout" from a LaserWriter. The fix for this error is to change the default timeout values in the LaserWriter itself, and this is accomplished by the following procedure: (1) Ensure that the Laser Prep File is sitting in the LaserWriter. This can be done by sending a small job to the LaserWriter (e.g., type "Hello" in MacWrite [see Note 1] and print it). (2) Type and send the following Postscript⁴ file to the LaserWriter:

```
%
% set job timeout for printer
serverdict begin 0 exitserver
statusdict begin 000 setdefaulttimeouts
```

One way to send the Postscript file is via the application SendPS.⁵ This will reset the timeout parameters and avoid errors. Having cleared this hurdle, there will be others, which are presently under investigation. Occasionally, "I/O Error in PICT File" is generated, but this can sometimes be rectified by running the program again and printing the image with a slight change in one or more of the view parameters. Finally, if the printing and screen display formats do not suit, the code for these can easily be changed in the subroutines "Three_D_Plotter," "Display_CFT," and "Display_Clusters."

APPENDIX C

A Formal Account of the k means Cluster Analysis

This formal description of the k means clustering technique applied to a CFT distribution is based on one given by Hunt (1975, pp. 98-99). In this context, the objects to be classified are the locations in the CFT distribution. Each location can be thought of as an object with three attributes: a row coordinate, a column coordinate, and a cumulative fixation time or t value.

Let r refer to the row coordinates of the CFT distribution, where $r = 1, 2, \dots, r$; r = total number of rows. Let c refer to the column coordinates of the CFT distribution, where $c = 1, 2, \dots, c$; c = total number of columns. Let t refer to the t values in the CFT distribution, where $t = 1, 2, \dots, t$; t = total number of cumulative fixation times. Therefore, $t = r \cdot c$.

Thus, for the moment, each object can be regarded as a triple (trc), which, when quantified, provides a complete description of that object. For example, the triple (25:12:49) identifies t value 25 residing at row 12 and column 49 in the CFT distribution. Let the triples in the CFT distribution be ordered into the sequence S_{trc} consisting of $(trc)^1, (trc)^2, \dots, (trc)^n$ in a two-dimensional description space.

Given these preliminary definitions, it is assumed that exactly k clusters exist within the CFT distribution and that each cluster has a mean point m_j^* . In the present context, this mean point is, of course, a mean geographical location within the CFT, that is, the center of gravity of cluster j . This mean point reduces, in fact, to a mean triple, $(trc)_j$, that is, the triple describing the central gravitational point of cluster j . What we require is an algorithm that estimates $\{m_j^*\}$ based on S_{trc} .

MacQueen's Algorithm

Applied to a CFT distribution, MacQueen's (1967) algorithm is performed as follows:

1. Select a set of initial estimates, $M^0 = \{m_i^0\}$, by assigning each of the k highest and most isolated values of t in S_{trc} to a distinct cluster. At this point, $m_i^0 = (trc)^i$, where $i = 1, 2, \dots, k$.

2. When a new triple $(trc)^i$ is selected at random from S_{trc} , and where $(trc)^i$ is not a member of the set $\{m_j^0\}$, assign the new triple to the cluster j , for which the Euclidean distance $d\{m_j^0, (trc)^i\}$ is minimized. Ties are resolved in favor of the lowest indexed cluster.

3. Calculate the new mean estimates, $\{m_j^{i+1}\}$ by computing the means of the points in each cluster.

3a. If $(trc)^i$ was not assigned to cluster j , m_j^i is unchanged.

3b. If $(trc)^i$ was assigned to cluster j , the new mean estimate is based on the current items in the cluster.

4. The calculation of the new cluster mean m_j^{i+1} , when the triple has been assigned, falls into two separate calculations, one for the new cluster row mean and one for the new cluster column mean. Calculating the new row mean first: Let w_j^i be the number of triples already assigned to cluster j at the time the i th triple is presented. Let $(.r.)^i$ be the row value of triple $(trc)^i$ being presented. Let $m^i(.r.)_j$ be the current mean row for cluster j . Finally, let $m^{i+1}(.r.)_j$ be the new mean row for cluster j . Then

$$m^{i+1}(.r.)_j = \left[\frac{1}{w_j^i + 1} \right] \{w_j^i \cdot m^i(.r.)_j + (.r.)^i\} \quad (1)$$

The new column mean for cluster j is calculated in a similar manner: Let $(.c)^i$ be the column value of triple $(trc)^i$ being presented. Let $m^i(.c)_j$ be the current mean column for cluster j . Let $m^{i+1}(.c)_j$ be the new mean column for cluster j . Then

$$m^{i+1}(.c)_j = \left[\frac{1}{w_j^i + 1} \right] \{w_j^i \cdot m^i(.c)_j + (.c)^i\} \quad (2)$$

Thus, the new row and column means determine the new center of gravity or m_j^i for cluster j . In other contexts, MacQueen (1967) proved that if the sequence of data points or objects to be clustered is actually generated by samples from k different probability distributions, then the k estimates m_j^i will asymptot-

ically converge on the means of the distributions, and this is also the case with the CFT distribution. Furthermore, the CFT case allows for several alternatives in the computation of k means. I shall outline one of these alternatives.

CFT Weighting

Instead of using the denominator $w_j^i + 1$ (the number of points or triples already assigned to the cluster plus one) in Formulas 1 and 2, it is possible to substitute as denominator the sum of all the t values of triples in cluster j at the time the i th triple is presented plus the t value (cumulative fixation time) of the triple $(trc)^i$ currently being assigned to cluster j . Let T_j^i = the sum of all the t values of triples in cluster j at the time the i th triple is presented. Let $(t.)^i$ = the t value (cumulative fixation time) of the triple $(trc)^i$ currently being assigned to cluster j . The new denominator then becomes $T_j^i + (t.)^i$.

It is, of course, also necessary to add this term to the row or column value appearing within the braces of the formula. This method weights the position of the new mean m^{i+1} by the total cumulative fixation time of triples already assigned to cluster j plus the total fixation time of the new triple being assigned to cluster j . The formula for computing the new mean row then becomes

$$m^{i+1}(.r.)_j = \left[\frac{1}{T_j^i + (t.)^i} \right] \{T_j^i \cdot m^i(.r.)_j + (t.)^i \cdot (.r.)^i\} \quad (3)$$

The formula for computing the new column mean is

$$m^{i+1}(.c)_j = \left[\frac{1}{T_j^i + (t.)^i} \right] \{T_j^i \cdot m^i(.c)_j + (t.)^i \cdot (.c)^i\} \quad (4)$$

(Manuscript received January 13, 1988;
revision accepted for publication August 13, 1988.)