Universität Potsdam

Reinhold M. Kliegl

EMAN : a modular and iterative eye-movement analysis program

first published in:

Theoretical and applied aspects of eye movement research : selected ed.
proceedings of The Second European Conference on Eye Movements,
Nottingham, England, 19 - 23 Sept., 1983 / ed. by Alastair G Gale and Frank
Johnson. - Amsterdam u.a. : Elsevier, 1984. - (Advances in psychology ; 22),
S. 31-37

Postprint published at the Institutional Repository of the Potsdam University:

In: Postprints der Universität Potsdam

Humanwissenschaftliche Reihe ; 141

<http://opus.kobv.de/ubp/volltexte/2009/3990/>

<http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-39909>

Postprints der Universität Potsdam

Humanwissenschaftliche Reihe ; 141

EMAN: A MODULAR AND ITERATIVE EYE-MOVEMENT ANALYSIS PROGRAM

Reinhold M. Kliegl

Max Planck Institute for Human Development
and Education
Berlin, West Germany

EMAN is an eye-movement analysis program that consists of four modules. The first module rescales eye positions to coordinates of the display. The second and third modules reduce data to a fixation format and identify areas of bad measurement by means of iterative passes over the data. In the fourth module iterative algorithms are employed for the identification of line numbers and for achieving congruence between fixations and display.

INTRODUCTION

Videobased systems for monitoring eye movements - such as the Applied Science Models or Demel's DEBIC 80 - generate information about eye position and pupil diameter at 60 or 50 Hz. The program EMAN was written for the analysis of this eye-position information. EMAN first rescales machine output to coordinates of the display as described in Kliegl and Olson (1981). The rescaled values are then processed by three other program modules that are reported in this paper. First, reduction of valid samples to position clusters and reduction of low-data-quality samples to blink clusters is accomplished in the Module REDUCE. The second module, CLEANUP, performs additional checks on the data-quality of position clusters and removes saccadic position clusters from the data base. Finally, the output of CLEANUP, a sequence of fixations, is matched to the text display by Module TEXT. This module permits interactive changes of parameters to achieve congruence between fixations and lines of the text display. Of course, the last module is only meaningful for the analysis of eye movements that were collected during reading.

The program was parameterized on the basis of the following three assumptions: One, the machine resolution or the display size allow reliable identification of fixations at the level of letters. Two, with a 60 Hz sampling rate, only clusters encompassing at least six samples (i.e. lasting longer than 100 ms) will be called a fixation. Three, clusters with fewer samples may occur between fixations and indicate the time necessary to shift the eye. Since it takes up to six samples to complete a return sweep across a line of text of about 20 degrees of visual angle, the time between two fixations within a line should not exceed five samples:

In programming EMAN it was not assumed that data obtained during a certain trial are of equally good quality. To the contrary, each program module was motivated by a different set of problems all or some of which may apply during a given trial. In the description of the modules, emphasis will be placed on the problems they address and how solutions were implemented. In

general, conservative solutions are advocated since eye-movement researchers are usually blessed with enough data to compensate for segments of low-quality data.

MODULE REDUCE

Prior to reduction there must be a criterion to distinguish clearly invalid from potentially valid measurements. In videobased systems invalid measurement most obviously occurs when the subject blinks or when the identification of pupil and corneal reflexion fails. Both conditions lead to a zero or greatly reduced pupil diameter. To determine a criterion for valid measurement the program computes mean and standard deviation of non-zero pupil diameters. Based on either a multiple of standard deviations or a certain percentage of the mean, a lower boundary for pupil diameter is set. After determining the criterion for potentially valid measurements, the data are processed by the algorithm displayed in Figure 1. If the pupil diameter is below the criterion, the sample is called a BLINK. Successive blinks are collapsed into a BLINK CLUSTER. A sequence of valid samples is combined to a POSITION CLUSTER if their coordinates are sufficiently similar. Note, however, that some clusters may contain only one sample.

```

:
:
BLINK?

    YES:      WAS LAST SAMPLE (CLUSTER) BLINK?

              YES:  UPDATE BLINK CLUSTER

              NO:   FINISH POSITION CLUSTER
                   START NEW BLINK CLUSTER

    NO:       WAS LAST SAMPLE (CLUSTER) BLINK?

              YES:  FINISH BLINK CLUSTER
                   START NEW POSITION CLUSTER

              NO:   IS PRESENT SAMPLE (CLUSTER) WITHIN WINDOW?

                   YES:  UPDATE POSITION CLUSTER

                   NO:   FINISH POSITION CLUSTER
                        START NEW POSITION CLUSTER

:
:

```

FIGURE 1. Reduction algorithm in Module REDUCE

If the pupil diameter is below the criterion, the top block of the algorithm is executed. If the last sample was a blink as well, the present sample is simply added to the blink cluster. If the last sample was valid, then the previous position cluster is closed and a blink cluster is started.

If a sample passes the first validity test, it will enter the second block of the algorithm. There, if the last sample was a blink, the blink cluster

is closed and, since positions during blinks are not valid, its x- and y-values are set to zero. The present sample then becomes the seed for a new cluster of valid positions. If the last sample was valid as well, then, if the coordinates of the present sample are within a window around coordinates based on the running mean of a cluster of previous samples, the present sample is added to this cluster. Otherwise, the previous cluster is finished and the present sample becomes the seed for a new cluster. The algorithm that decides whether a sample is within a window was suggested by Mason (1976). A detailed description of the present adaptation can be found in Kliegl and Olson (1981). Since the positions were rescaled to display coordinates and since the resolution of the eye-tracker under the viewing conditions in the laboratory were plus or minus one character, the minimal distance between clusters was set at one.

New position clusters are formed if a single sample falls outside the window. The cluster resulting from this seed, however, may be within the window of the preceding position cluster. To ensure that successive clusters are at least the minimum distance apart from each other, the clusters themselves are processed by the reduction algorithm. Successive clusters are collapsed into one, if they are closer than the minimum distance. The algorithm is applied iteratively until no further reduction in the number of clusters occurs. Thus, the reduction iteration will stop, once all neighbouring position clusters have a minimum distance from each other.

Experience has shown that occasionally two clusters with virtually identical positions were separated by a single cluster containing mostly one or two samples. Since it is impossible that an eye movement can leave and come back to a certain position without an intervening fixation, it is assumed that these samples reflect spurious measurement and that the two clusters are actually one. To identify and remove outliers of this kind a second iterative algorithm compares triplets of neighbouring clusters. The outlier is counted for the duration, but its position information is ignored for the computation of cluster coordinates. The iteration stops once no further reduction in number of clusters occurs.

At the end of this two-stage iterative reduction algorithm the data are available as a sequence of clusters. For position clusters we know how many samples they encompass, their average horizontal and vertical positions and their average pupil diameters. Blink clusters have zero values in position and pupil diameter vectors. So far no fixation-saccade distinction was made and no sample was removed from the data base. The next module, CLEANUP, serves this purpose.

MODULE CLEANUP

The criterion for distinguishing valid from invalid measurements alone is not sufficient. There are other conditions that yield doubtful data quality, but are not necessarily reflected in loss of, or sufficiently reduced values of pupil diameters. For example, prior to a blink or loss of delimiters, the position information is already contaminated by the ensuing loss of measurement. Analogously, low data quality may be observed after an unstable measurement situation. To ensure data quality around blink clusters, the algorithm shown in Figure 2 is used. The data base is searched for blink clusters. Once a blink cluster is encountered, the program engages in a backward search until two consecutive position clusters with similar vertical position are found. Position clusters too discrep-

ant from the previous ones are recoded as blink clusters. A similar algorithm is used for a forward search after a blink cluster.

```

:
:
IS CLUSTER N A BLINK CLUSTER?

NO: N: = N + 1

YES: NI: = N
    FLAG IS "FALSE"
    UNTIL FLAG IS TRUE: ITERATE

ITERATE: HAVE POSITION CLUSTERS (NI - 1) AND (NI - 2) SIMILAR
          VERTICAL POSITION?

YES: FLAG IS "TRUE"
    N = N + 1

NO: RECODE POSITION CLUSTER (NI - 1) AS BLINK CLUSTER
    NI = NI - 1

END OF ITERATE
:
:

```

FIGURE 2. Ensuring stable measurement around blink clusters

In this case, only vertical information is used since unstable measurement mainly occurs at the end or beginning of lines. At these locations horizontal positions are of course maximally discrepant. Since unstable measurement usually results in a big variation of vertical positions, the similarity between two consecutive position clusters can be set fairly liberal, e.g., at two or three lines in a reading situation.

The next step towards a clean sequence of fixations is to remove blink and saccadic position clusters (i.e. clusters with fewer than six samples) from the data base. While deleting these clusters, the program records how many samples are removed between two fixations. This permits the identification of areas of low data quality that never resulted in an actual loss of delimiters. Usually a sequence of position clusters with few samples per cluster is indicative of a condition in which the time between two fixations might well exceed estimates of durations of saccadic movements. As mentioned previously, the maximum time between fixations within a line should be shorter than the time for a linesweep. If the time is longer than five samples, it may be inferred that the two fixations delimit an area of bad measurement.

To retain information about areas of blink clusters and about areas of bad measurement, the fixations delimiting these conditions are flagged in the vector coding the pupil diameter. The fixation preceding the bad area contains the number of samples that were lost; the fixation following the bad area codes whether it was a blink cluster or unstable measurement that caused the loss. This information is quite important in future analyses

since it prevents wrong inferences about information processing between the two fixations. For example, we most certainly must not conclude that this area was skipped with a large saccade.

MODULE TEXT

For eye movements collected during reading, the described data processing was often not sufficient to achieve congruence between fixations and the appropriate lines of the text paragraph. Reasons for the failure could have been a temporarily invalid calibration table, greater machine noise associated with the vertical position, or a subject's fixation habits. In general, even the presence of only a small systematic drift may cause a great similarity in vertical positions of fixations at opposite ends of successive lines and may lead to misplaced fixations.

The TEXT module was written to remove systematic drift in vertical position within a line (see Figure 3). The program compares successive pairs of fixations. If their vertical and horizontal positions are within a certain range they are considered to be from the same line. In this case, if their vertical positions are not close to identical, the vertical position of the first fixation is replaced with the average of both. An uninterrupted sequence of fixations judged to be within the same line will be called a cluster of fixations within a line (FWL). The program iteratively processes the data base until all fixations within FWL clusters have virtually identical positions.

```

:
:
FLAG IS "TRUE"

WHILE FLAG IS "TRUE" DO:
    FLAG IS "FALSE"

    FOR N = 1 TO (NUMBER OF FIX -1) DO:
        ARE NI AND (N + 1) FIXATION ON THE SAME LINE?

            YES: ARE VERTICAL POSITIONS IDENTICAL?

                YES: N = N + 1

                NO: VERT. POS. OF FIX. N = AVERAGE OF VERT. POS.
                    OF FIX. N AND FIX. (N + 1)
                    FLAG = "TRUE"
                    N = N + 1

            NO: N = N + 1

    END OF FOR LOOP

END OF WHILE LOOP
:
:

```

FIGURE 3. Determining vertical position for fixations within a line

Once the iteration stops it is likely that more FWL clusters were identified than the text contains lines. If two fixations within a line were separated by a large saccade or if the subject jumped back to re-read part of the text, sequences of fixations within one line will yield separate FWL clusters. To arrive at the appropriate set, FWL clusters indicating the same lines need to be collapsed.

Under default parameterization the program assumes a fixed minimum distance between "true" lines and combines FWL clusters that are within this margin. Ideally, the number of lines identified by the program should correspond to the number of lines the subject read. An immediate evaluation of the quality of the computations is possible by inspecting the plot of fixations and text which the program generates on the terminal. From this plot it is immediately obvious if the program found too many or too few lines. If the plot is bad, the user has the option of telling the program the number of lines it should find. This will invoke another iterative procedure. At the beginning, the minimum distance between lines is zero, but will be slowly increased with every iteration until the distance has been identified that generates the desired number of lines. Again, a terminal plot may be used for evaluative purposes. If the plot is still not adequate, the user has the option of changing parameters that determine whether fixation pairs should be allocated to a FWL cluster. A change in these parameters will result in a different set of FWL clusters and, consequently, in different line values. If the program fails again, one should probably delete parts of the text for which congruence could not be achieved - or improve on the algorithms proposed in this paper.

TECHNICAL INFORMATION

EMAN was written in VAX/VMS Fortran 77 and is implemented on DEC VAX 11/780 machines. Input to the program are x,y-coordinates and pupil diameters collected with a fixed sampling rate. The program reads all the records of a trial into three vectors. Two additional vectors store the number of samples in a cluster and the record number of the first sample in the cluster. At the beginning, the first vector is a unit vector; the second vector simply contains the record numbers. These five vectors form a workspace matrix common to all modules. All computations are carried out on workspace elements with results replacing original values. In Module RE-SCALE x and y coordinates in machine units are replaced with x and y coordinates of display dimensions. Modules REDUCE and CLEANUP effect elements of all vectors. Whenever samples are combined in a cluster, the number of samples and their average position and pupil diameter are stored in corresponding elements of the first sample of the cluster. Then, the samples following the cluster are shifted up the appropriate number of elements. Thus, with every iteration the length of the workspace vectors is reduced. It is possible to obtain output after every iteration to monitor the reduction and "cleaning" process. After data were processed in TEXT, the workspace vectors contain number of samples, horizontal display position, line number, pupil diameter (or a flag indicating loss or unstable measurement following/preceding the fixation), and the record number of the first sample of this fixation. The latter permits immediate identification of the fixation in the raw data. The output is suitable for input to programs FIXPLT and MATCH described in Kliegl (1981). Substitution of the TEXT module with algorithms designed for analyses of other than paragraph-reading data is possible. Modules available are for analyses of tracking-alight task (Pavlidis (1981), Olson, Kliegl, & Davidson (in press)) and

the analyses of single-word reading data. Despite the large number of iterative procedures the analysis is quite fast. For example, an analysis of eye movements collected during reading of a 20-line long text paragraph (6538 records, reduced to 354 fixations) was accomplished within one minute. The program is available upon request.

REFERENCES

- Kliegl, R., Automated and interactive analysis of eye fixation data in reading, *Behavior Research Methods and Instrumentation* 13 (1981) 115-120.
- Kliegl, R. and Olson, R. K., Reduction and calibration of eye monitor data, *Behavior Research Methods and Instrumentation* 13 (1981) 107-111.
- Mason, R. L., Digital computer estimation of eye fixation, *Behavior Research Methods and Instrumentation* 8 (1976) 185-188.
- Olson, R. K., Kliegl, R. and Davidson, B. J., Dyslexic and normal reader's eye movements, *Journal of Experimental Psychology: Human Perception and Performance* (in press).
- Pavlidis, G. Th., Do eye movements hold the key to dyslexia?, *Neuropsychologia* 19 (1981) 57-64.