EyeMap: a software system for visualizing and analyzing eye movement data in reading

Siliang Tang · Ronan G. Reilly · Christian Vorstius

Published online: 13 October 2011 © Psychonomic Society, Inc. 2011

Abstract We have developed EyeMap, a freely available software system for visualizing and analyzing eye movement data specifically in the area of reading research. As compared with similar systems, including commercial ones, EyeMap has more advanced features for text stimulus presentation, interest area extraction, eve movement data visualization, and experimental variable calculation. It is unique in supporting binocular data analysis for unicode, proportional, and nonproportional fonts and spaced and unspaced scripts. Consequently, it is well suited for research on a wide range of writing systems. To date, it has been used with English, German, Thai, Korean, and Chinese. EyeMap is platform independent and can also work on mobile devices. An important contribution of the EyeMap project is a device-independent XML data format for describing data from a wide range of reading experiments. An online version of EyeMap allows researchers to analyze and visualize reading data through a standard Web browser. This facility could, for example, serve as a front-end for online eye movement data corpora.

Keywords Eye movements · Reading · Data visualization

The latest version of EyeMap can be retrieved from sourceforge.net/projects/openeyemap/files. The online system is available at http://eyemaponline.tk.

S. Tang · R. G. Reilly ()
Department of Computer Science, NUI Maynooth,
Maynooth, Co.Kildare, Ireland
e-mail: ronan.reilly@nuim.ie

C. Vorstius

Department of Psychology and Florida Center for Reading Research, Florida State University, Tallahassee, FL, USA



Over the last 2 decades, one of the most successful tools in the study of human perception and cognition has been the measurement and analysis of eye movements (for recent overviews, see Duchowski, 2002; Hyöna, Radach, & Deubel, 2003; van Gompel, Fischer, Murray, & Hill, 2007). An area that has particularly profited from recent technical and methodological advances is research on reading. It has become a central field in cognitive science (Radach & Kennedy, 2004; Rayner, 1998) and has included the development of advanced computational models based on eye movement data (e.g., Engbert, Nuthmann, Richter, & Kliegl, 2005; Reichle, Pollatsek, Fisher, & Rayner, 1998; Reilly & Radach, 2006). At the same time, eye tracking during reading is also beginning to provide a major arena for applied research in fields as diverse as clinical neuroscience (Schattka, Radach, & Huber, 2010), training (Lehtimaki & Reilly, 2005), and elementary school education (Radach, Schmitten, Glover, & Huestegge, 2009).

Eye-tracking techniques have gradually improved during the last few decades. Many commercial trackers have emerged that use video oculography, which is a noninvasive and relatively accurate eye-tracking technology. However, many trackers are general purpose and have either low spatial accuracy (greater than 0.2°) or a low sampling rate (less than 250 Hz), which makes them unsuitable for modern reading research. A few, such as the SR Research EyeLink series (http://www.sr-research.com) and the IVIEW X series from SMI (http://www.smivision.com), have sufficient spatial resolution to support letter-level spatial accuracy, while at the same time providing sufficient temporal resolution for millisecond-level viewing time analyses (see http://www.eyemovementresearch.com/eye%20trackers.htm for a comprehensive overview of available hardware).

A similar situation pertains for eye movement data analysis software in reading. The main manufacturers

provide proprietary software that delivers good solutions for a range of data analysis tasks. Typical examples are the Data ViewerTM software from SR Research, the BeGazeTM Analysis Software from Sensomotoric Instruments (SMI), and the Studio Software SuiteTM from Tobii (http://www.tobii.com). Programs like these are integrated with system-specific software packages for stimulus presentation and measurement control. They offer a range of options that are very useful for many applications. However, these currently available software options also have a number of substantial limitations:

- They tend to have few functions for processing reading data and generating useful variables (measurements on eye movement events in specific areas of interest, such as letters and words).
- 2. Most have limited options for text presentation and analysis. Some cannot load text material (users have first to convert the text into an image). Some do not support Unicode and, therefore, have problems dealing with writing systems other than Roman-based ones. Some do not readily support the editing of eye movement data. For example, the otherwise very advanced Data Viewer software by SR Research supports only manual drift correction for each text line and sometimes has problems identifying word boundaries.
- The software is hardware specific, and the data are stored in a proprietary format. Consequently, users cannot work across platforms, which causes problems for data sharing and result comparison.
- 4. They are not freely available, are difficult to extend, have limited visualization capabilities, and can handle only a limited range of experimental paradigms. As a result, researchers often have to develop their own analysis software for specific experiments.
- Most solutions are commercial software packages, and purchase and support can be quite expensive for academic researchers.

In order to resolve the above-mentioned problems and produce a more effective tool that allows psychologists to gain more insight into their eye movement reading data, we developed EyeMap, a freely available software system for visualizing and analyzing eye movement data specifically for reading research.

EyeMap supports dynamical calculation and exporting of more than 100 different types of reading-related variables (see Appendix 1 for the current list of variables). It allows researchers to load reading materials from an HTML file, which is convenient for creating, presenting, modifying, and organizing different experimental text materials. Furthermore, Unicode text can also be handled, which means that all writing systems, even unspaced scripts

such as Thai and Chinese, can be presented and segmented properly. Moreover, as far as we are aware, it is the first data analysis software that can automatically and reliably find word boundaries for proportional and nonproportional fonts. During the analysis, these boundaries are marked as areas of interest (AoIs) automatically. Another advantage of EyeMap is that it can work on a wide range of platforms, even mobile devices. It also contributes an XML (eXtensible Markup Language) based device-independent data format for structuring eye movement reading data. Another important contribution is that EyeMap provides a powerful user interface for visualizing binocular eve movement data. Direct data manipulation is also supported in the visualization mode. Furthermore, an EyeMap online version allows users to analyze their eye movement data through a standard Web browser, which is in keeping with the trend toward cloud computing environments.

Figure 1 presents all of the functional modules and associated data files involved in the EyeMap environment. In Fig. 2, we present an informal use-case analysis of the process of going from the binary formatted eyetracker data (top of figure) through to an exported CSV file (bottom of figure).

The gray rectangles in Fig. 2 represent processes carried out by EyeMap, the yellow ones represent EyeMapgenerated data, and the green ones represent user-supplied data sources. Note that in the current version of the system, the Data Parser is a stand-alone module, separate from the main EyeMap software. This permits its replacement when data files from different manufacturers are parsed.

It should also be noted that currently, EyeMap uses the results from the online saccade/fixation detection mechanism used by the specific eyetracker manufacturer. Therefore, differences in the often undocumented event detection algorithms used by different manufacturers currently carry over to EyeMap analyses. However, in principle, it should be possible to use a potentially more accurate postprocessing algorithm on the raw eye position samples extracted from the eye tracking record.

Isomorphic with the sequence of actions illustrated in Fig. 2, the workflow in EyeMap follows roughly four phases: data conversion, data integration, data visualization and editing, and data export and sharing. In the following sections, we will provide some more detail of these different phases.

Eye movement data conversion

Eye movement data are organized in many different ways, and each tracker manufacturer has its own data descriptions. In high-speed online trackers, which are deployed in most reading experiments, the raw data are stored in binary format, due to the requirement for fast data retrieval and storage. The lack of standardization of data format,



System Overview Saccade Playback **Fixation** Viewer Viewer raw data Heatmap Viewer edit mode Binary Scan Path Viewer Data convert Editor eye voice Raw Info. edit text material data Viewer formatted data raw data convert load EyeMap **ASCII** XML Raw Data **Events** variable export Viewer trial raw data Data Parser namespace variable list expt. config edit export.xml Editor word Legend csv output Variable properties trial attach export fixation/word properties Calculation reports Function Module Data Files

Fig. 1 System overview

however, can be an impediment to data exchange between different research groups. Some tracker manufacturers provide APIs or tools to export the binary data into ASCII, which is a convenient general format for data sharing but not sufficient for data processing, since the data in this format lack any structural or relational information. Some researchers have tried to develop device-independent data formats for a range of trackers. The first public device-independent data format was developed by Halverson and his colleagues (Halverson & Hornof, 2002). It has an XML-like structure, in which information such as experiment description, AoI information, and internal/external events are stored together. XML enforces a more formal, stricter data format that facilitates further data processing. Another XML-based generic data storage format was published as a deliverable from the EU-funded project COGAIN (Bates, Istance, & Spakov, 2005), which has become the de facto standard for gazebased communication systems. However, since it is designed for gaze-based communication, it focuses mainly on hardware aspects so as to be compliant with as many trackers as possible, which is not necessarily a helpful option for reading research, because only a limited number of types of trackers can be used in modern reading research (see above). On the other hand, reading researchers are more interested in the cognitive processing occurring during fixation/saccade/blink/pupil events, which is at a greater level of detail than the gaze-level analyses afforded by relatively low-temporal-resolution trackers. Therefore,

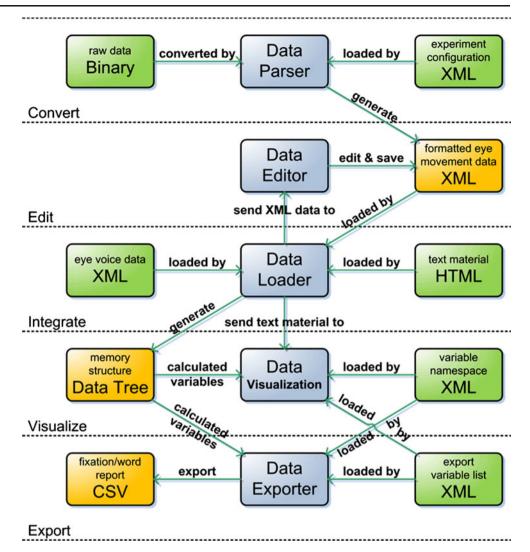
COGAIN carries with it a considerable overhead, while still not being specific enough for reading research. For these reasons, we developed a novel XML-based device-independent data format designed specifically for reading experiments to facilitate the calculation of eye movement measures that are specific to reading.

A reading experiment is always trial based, and on each trial, subjects are required to read some predetermined text, usually a sentence or paragraph per screen. To describe a reading experiment with all the related events that can occur during the experiment, we need at least the following elements in our XML description (see Fig. 3 for a sample structure):

- 1. <root> </root>: All the information of the experiment is inside the root pair. It contains < trial > elements. And each data file has only one < root > block.
- <trial> </trial>: Trial block contains all the information
 for a single experimental trial. Each block has <fix>
 and <sacc> elements. Each trial block has a unique
 number called "id," which starts from 0, indicating the
 number of the trial in the current experiment. Each trial
 also has an <invalid> tag, which represents whether the
 current trial is valid or not.
- <fix> </fix>: Fix block contains all the information for a fixation; each fixation block may have the following tags:
 - a. <eye>: the eye with which the current fixation event is associated;
 - b. <st>: start time-stamp in milliseconds;



Fig. 2 Use-case analysis of EyeMap



- c. <et>: end time-stamp in milliseconds;
- d. <dur>: duration (in milliseconds) of the current fixation;
- e. $\langle x \rangle / \langle y \rangle$: average x-, y-coordinates of the current fixation;
- f. <pupil>: average pupil size (diameter) in arbitrary units for the current fixation;
- g. <bli>g. <bli>duration (in milliseconds) of a blink if it occurred prior to the current fixation;
- h. <raw> *: the distance in pixels of the start (sx; sy) and end (ex; ey) x-, y-coordinates from the average x-, y-coordinates of the current fixation;
- i. <id>: a unique number starting from 0 (based on temporal order, where even numbers are used for the right eye and odd for the left eye);
- 4. <sacc> block contains all the information for a saccade; each saccade block will have the following tags:
 - a. <eye>: the eye with which the current fixation event is associated;

- b. <st>: start time stamp in milliseconds;
- c. <et>: end time stamp in milliseconds;
- d. <dur>: duration (in milliseconds) of the fixation;
- e. $\langle x \rangle / \langle y \rangle$: x-, y-coordinates of the start position;
- f. $\langle tx \rangle / \langle ty \rangle$: x-, y-coordinates of the end position;
- g. <ampl>: the total visual angle traversed by the saccade, reported by <ampl>, which can be divided by (<dur>/1000) to obtain the average velocity;
- h. <pv>: the peak values of gaze velocity in visual degrees per second;
- i. <id>: a unique number starting from 0 (based on temporal order, even numbers for right eye, odd for left eye).

With respect to the odd/even ID coding scheme, this has proved especially useful for helping to synchronize data from the right and left eyes for studies of binocularity effects. The SR DataViewer, for example, provides separate data exports for the right and left eyes, but the task of synchronizing these data streams is nontrivial. The use of



Fig. 3 An example of eye movement data XML

```
<root>
 <trial id="0">
    <fix>
      <eve>R</eve>
      \langle st > 927307 \langle /st \rangle
      <et>927470</et>
      <dur>164</dur>
      < x > 100.1 < / x >
      < y > 104.7 < / y >
      <pupil>1501</pupil>
      <raw sx="-2.7" sy="0.9" ex="2.4" ey="-0.4"/>
      <id>0</id>
    </fix>
    <sacc>
      <eye>R</eye>
      <st>927471</st>
      <et>927515</et>
      <dur>45</dur>
      < x > 102.6 < / x >
      < y > 104.1 < / y >
      < tx > 508.6 < /tx >
      < ty > 133.3 < /ty >
      <ampl>11.66</ampl>
      <id>0</id>
    </sacc>
 </trial>
</root>
```

odd/even ID numbers makes the synchronization process a lot more manageable.

A data parser was developed to convert different data sources into the aforementioned XML format. The data parser is actually a small lexical analyzer generated by Lex and Yacc (Johnson, 1978; Lesk & Schmidt, 1975; Levine, Mason, & Brown, 1992). The parser works on lists of regular expressions that can identify useful components from the ASCII data file, recognize them as tokens, and then recombine the tokens into XML elements. Currently, the parser works on EyeLink (http://www.sr-research.com) and Tobii (http://www.tobii.com) exported ASCII data formats. However, it would be quite straightforward to create a new parser for other eyetracking devices, since users need only to create a new list of regular expressions based on the output specification of the new device.

A data structure for reading

In a typical reading experiment, parameters of interest are either properties of the stimulus materials (words, characters, and their linguistic features) or measures of the ongoing oculomotor behavior during the dynamic process of acquiring this linguistic information. This produces two different perspectives on the experimental data: the text perspective and the event perspective. Moreover, reading researchers are usually interested only in some specific variables (or attributes) in any given experiment. Take some common variables in a text-reading experiment as an example: The variable launch distance is an event-based variable, which describes the distance in character spaces from the launch site of an incoming saccade to the beginning of the current word. In contrast, the variable gaze duration, defined as the sum of all fixations on the target prior to any fixation on another word, is considered a text-based variable in this classsification. In order to fulfill the data modification and variable calculation requirements of these two perspectives, the experimental data are organized as a tree structure containing six different levels (from the root to the leaves: experimental level, trial level, word level, gaze level, fixation level, and saccade level). The basic component of the experimental data tree is an abstract node class. The node class has the basic functions of constructing and



maintaining the relationships among the elements of the data tree, which is the key to all the variable calculations. For a node $N_{p'}$, if its *i*th child node is N_i , then the node N_i can be formalized as the following set:

$$N_i = \{i, i', N_p, N_{i-1}, N_{i+1}, A_{i_j} = \{A_{i_1}, A_{i_2}, \cdots, A_{i_j}\}, T_{i_n} = (N_{i_1}, N_{i_2}, \cdots, N_{i_n})\}$$

$$(N_i \in T_{p_n} \text{ and } N_i = N_{p_i})$$

As is shown in the formula above, the node N_i satisfies the following properties:

- 1. N_i has an ID that indicates that it is the *i*th child of its current parent.
- 2. N_i has another ID i' that indicates that it is the ith child of its old parent. If $i \neq i'$, this means that the current parent node N_p is not the original parent of N_i . In other words, node N_i is created by another node but adopted by N_p . Since the node class allows multiple parents to connect to one node, the ID i' can help a parent node to identify itself.
- 3. N_i has a reference to its current parent node N_p .
- 4. N_i has a reference to its predecessor (N_{i-1}) and successor (N_{i+1}) .
- 5. N_i has a set of attributes (A_{ij}) , which contains the specific information of node N_i .
- 6. N_i contains a *n*-tuple T_{i_n} that records an ordered list of child nodes.

From the vertical perspective, the node class builds a bidirectional link between a parent node and its children, while from the horizontal perspective, the node class retains bidirectional references to children at the same level. The vertical perspective can simplify variable calculations. For example, once you reach the gaze node G_i , the gaze duration on G_i can simply be calculated by the following formula:

$$G_i imes A_{i_{ extit{gazeDur}}} = \sum_{j=1}^n G_{i_j} imes A_{i_{ extit{fixationDur}}}.$$

This effectively involves the traversal of the gaze subtree and the accumulation of fixation durations in the process. A horizontal perspective, on the other hand, is extremely useful in generating word-level data matrices for output and analysis. Typically, in a reading experiment, the analysis focuses, in turn, on the fixation event level and on the objects of experimental interest, usually the words and sentences of the text. EyeMap allows the export of separate event-level and word-level data matrices.

Figure 4 is a fragment of a dynamic memory data structure created for processing a trial of a single line of text-reading data. Inside the green box, the word *stinking* has three right-eye fixations, and these fixations belong to

two different gazes. Thus, as is depicted in Fig. 4, the WORD5 node contains all the information (word boundary, word center, word frequency, etc.) for the word *stinking*. It has two child GAZE nodes. Three FIXATION nodes are allocated to two GAZE nodes on the basis of the gaze definition.

To optimize the loading speed, the data tree is not created at one time. Each time that a trial is presented/modified, the corresponding trial branch is created and/or updated; thus, the information in the visualization is always up to date.

Data integration

In the current implementation of EyeMap, there are two sources of data that can be used to augment the eye movement data. A mandatory data source is the file detailing the text being read (i.e., text.html). Clearly, any eye movement data are uninterpretable without information about the words being viewed. The text information is provided in a HTML formatted file that also provides font type and size information. The latter plays an important role in word and letter segmentation.

An additional, optional facility is to synchronize speech data with eye movement data. Voice data are incorporated in two ways: as part of the visualization system and as part of the data export facility.

For visualization, the articulation start time and duration are extracted from a user-supplied voice.xml file for each word. When the user examines a word on the current trial, EyeMap will play the associated recording of that word.

For data analysis, the articulation start time is synchronized with the experimental timeline. It is therefore possible to associate when word articulation starts or ends with which fixation was being made at that time and what word or letter was being fixated. In this way, EyeMap can readily provide information about the eye voice span.

Data visualization and editing

The main job of this phase is to represent the experimental stimuli and visualize the eye movement data in different



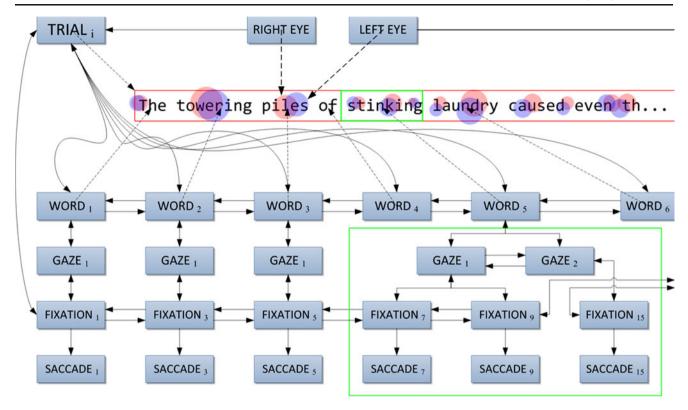


Fig. 4 Nodes in memory

ways, which facilitates the user in understanding and exploring the data from different perspectives. In a reading experiment, the primary stimulus is text. Figure 5 shows some sample experimental data using EyeMap with different combinations of Unicode, proportional, and non-proportional fonts and spaced and unspaced scripts. So far, EyeMap has been used for research on a wide range of writing systems, such as English, German, Thai, Korean, and Chinese.

As is shown in Fig. 5, EyeMap not only recreates the experimental stimulus, but also finds the word center (yellow line) and word boundary automatically, with and without punctuation (red vs. green boundary), and marks them as AoIs. As far as we are aware, EyeMap is the first software capable of reliably autosegmenting proportional fonts in this way. For example, the SR Data Viewer has problems handling proportional fonts. Figure 6 gives a sample comparison with EyeMap.

Fig. 5 Representing various texts with recorded eye movement data and word boundaries.

a) multiline spaced English text with proportional font and punctuation, b) single-line spaced Korean text with non-proportional font, c) single-line unspaced English text with non-proportional font, d) single-line unspaced Chinese text with non-proportional font



d) single- line unspaced Chinese text with non-proportional font



Fig. 6 Comparison of SR Data Viewer and EyeMap segmentation algorithms

SR Data Viewer

In just over a month the deadly disease infected two-thirds of the population.

EyeMap

In just over a month the deadly disease infected two-thirds of the population.

There appear to be several problems with SR Data Viewer's autosegmentation algorithm. In Fig. 6, it treats the two words "of the" as one. Performance deteriorates further when the algorithm has to work with colored texts. Also, the algorithm always draws the word boundary through the middle of the space between the two words, while reading researchers normally treat the space preceding a word as part of that word. Segmenting letters for other than monospaced fonts is also highly unreliable. EyeMap, on the other hand, can detect the boundary of every letter in the word, even for text materials comprising mixed proportional font types with letter kerning.

EyeMap achieves this superior performance by taking advantage of the Flex implementation platform. It provides an application programmer interface (API) to a wide range of functions for segmenting paragraphs into lines, lines into words, and words into characters. Flex does all this on the basis of font metrics, rather than the analysis of an image, as is the case with Data Viewer. Note, however, that for word segmentation in writing systems that do not explicitly indicate word boundaries, EyeMap must rely on a user-specified separator incorporated in the HTML text file.

EyeMap offers one-dimensional (1-D), two-dimensional (2-D), and dynamic data visualizations (replays), which allow for a wide range of visualization. These visualizations

are implemented as *viewers*, representing a conceptual model of a particular visualization, with supporting panels. There are six different viewers, four of which are 2-D viewers. These are the fixation viewer, saccade viewer, playback viewer, and pure scan path viewer. The rest are 1-D charts, which comprise the pupil size viewer and an events viewer.

An example of a 2-D viewer is the fixation viewer shown in Fig. 7. Its contents are dynamically changed according to the selected trial. It contains a text background that shows the stimulus page of text for the current trial. Fixations are represented by semitransparent circles with different colors indicating the two eyes. When the user clicks the corresponding object (fixation dots, saccades, or words) inside the viewer, all the properties of that object are displayed. For example, when the word "see" is selected in Fig. 7, the word-level variables of "see" are calculated automatically and presented on the word properties list right of screen. The viewers also allow quick access to object data through pop-up tips. When the mouse cursor moves over an object, a tip with relevant information about the underlying object appears close to it.

Although the 2-D viewers implemented in EyeMap have some features in common, they plot different information on the text stimulus. Figure 8 demonstrates six available

Fig. 7 A zoomed fixation viewer

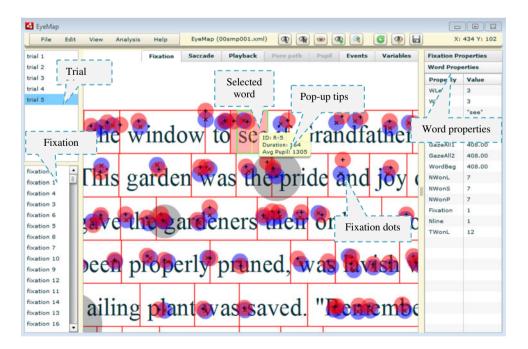
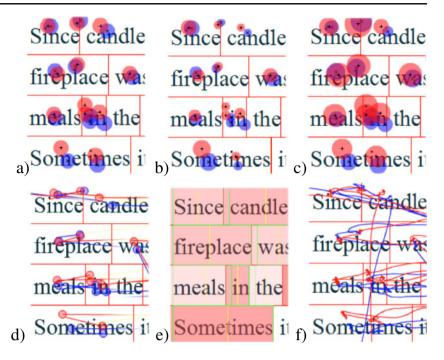




Fig. 8 Two-dimensional eye movement plot with a fixation dot with fixed circle radius, b fixation durations as the circle radius, c pupil size as the circle radius, d gradated colored saccade lines with gaze point, e word heatmap with word boundaries, and f pure scan path



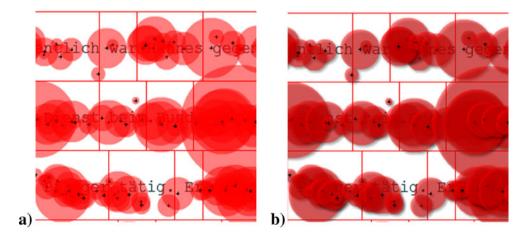
visualizations from the same text stimulus page. The snapshots a— c are taken from the fixation viewer. The fixation dot size in snapshots b and c corresponds to different measures, with Fig. 7b using fixation duration as the circle radius, while in Fig. 7c, the radius represents mean pupil size. Example d is taken from the saccade viewer, which connects gaze point (mean fixation position) with a gradually colored line indicating saccade distance and directions. Furthermore, snapshot e is a word heat map captured from a dynamic gaze replay animation in the playback viewer. Finally, snapshot f depicts a pure scan path representing real saccade trajectories.

EyeMap also utilizes different visual effects to enhance each viewer. For example, a drop-shadow effect can be activated in fixation viewers, which can be useful in some situations. For example, Fig. 9 shows some data collected from a patient with pure alexia, using a letter-by-letter reading strategy (unpublished data), where the drop-shadow very clearly indicates the temporal unfolding and spatial overlap of multiple fixations on the same word.

In the playback viewer, as can be seen from Fig. 10, when gaze animation is performed over the static text background, the semitransparent fixation dot has a "tail" to highlight the most recent fixations.

Time-correlated information, such as eye movement events, pupil dilation, mouse movements, keystrokes, stimuli changes, speech, and so forth, may be better represented in 1-D visualizations. Figures 11 and 12 give an example of pupil dilation versus time and eye movement events versus time in a 1-D chart, which we have implemented as a pupil viewer and an event viewer. In the event viewer, time range is controlled by a time axis bar; the user can control the observation time by moving the slider on the time axis or dragging the mouse around the chart.

Fig. 9 Fixation dots with/ without drop shadow





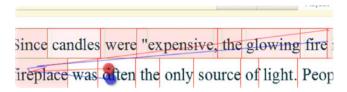


Fig. 10 Gaze playback with a virtual "tail"

In addition to visualizing reading data, direct data manipulation and modification through visualization are also supported by EyeMap. This function is extremely helpful in some cases, since fixation drift may occur during experimental trials despite good calibration, especially when working with children and patients. In a typical single-line reading experiment, a drift correction target dot may appear at the beginning of each trial, and the spatial reference grid for the calibration would be shifted accordingly. However, in a multiline text-reading experiment, as shown in Fig. 6, although the reading data are recorded by a relatively high-quality tracking system (e.g., EyeLink II at 500 Hz) with a drift correction before each trial, the data might not necessarily be clean, due to a small drift over time, especially in the vertical dimension. In such cases, manual drift correction can solve the problems of misalignment, when a few fixations cross over into the space assigned to the next line of text. As an illustration, Fig. 13 presents an example of how a near-ideal scan path can be produced with little effort in offline drift correction. Note that we recommend using this feature only for small corrections in real research studies so that natural variability of eye movement patterns is not distorted.

EyeMap offers a convenient edit mode, which allows users do correct fixation locations without having to use the keyboard. In edit mode, users can select multiple dots by dragging a rectangle and align them by selecting a corresponding option from the right-click mouse context menu. To move the selected fixation dots up and down as one group, users can use the mouse wheel directly. A double click on the view will simply submit all the intended modifications.

It is also possible to align globally, in the horizontal axis, all fixations with the drift correction dot. This is effective only for displays involving single lines of text. Its advantage is that it operates on a complete data set instantaneously. However, if the experiment involves multiline displays of text, alignment has to be done manually, as described in the previous paragraph.

Ultimately, the whole point of a system like EyeMap is to generate a data matrix from an experiment that can be statistically analyzed. EyeMap provides an easy-to-use interface for the selection of appropriate variables at the fixation event and word level. Once a standard set of variables has been decided upon, these can be stored in the export.xml file and will be automatically loaded in future sessions. As was already mentioned, EyeMap provides over 100 variables of relevance to the researcher (see Appendix 1 for the current list). The user can see the full list along with a brief description in EyeMap's variable viewer (see Fig. 14). The output data format is a CSV file with a header containing the exported variable names.

The fixation and word report output is dependent on the data tree module described in the data structure section. The exported variable names and their associated eye are listed in the user supplied *export.xml* file, which can also be created or modified in the EyeMap *export.xml* Editor (variables viewer). The *export.xml* file is loaded by EyeMap automatically. As is shown in Fig. 14, users can add variables to the export list by a drag-and-drop from the list of the available variables on the left. Variables can also be removed from the export list by similarly moving them to the Trash list on the right.

Data sharing is important in all lines of research. However, the current situation with oculomotor reading research is that most research groups compile their text materials and collect eye movement data for their own purposes, with limited data sharing. Large amounts of similar data are collected. In recent years, some research groups have started to create eye movement data corpora based on text-reading studies that have been made available to fellow researchers (Kennedy & Pynte, 2005; Kliegl, Nuthmann, & Engbert, 2006). A long-term goal should be to develop online databases for all the researchers to share freely their experimental setups, methodologies, and data. As a significant step toward this goal, we have created an online version of EyeMap at http://eyemaponline.tk. The

Fig. 11 Mean pupil size variation versus time

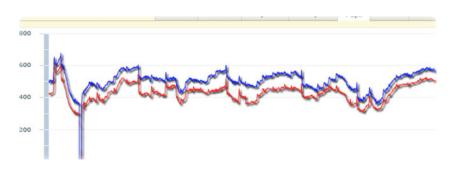




Fig. 12 Fixation events on words as a function of time



application is presented in Fig. 15 as running within a Chrome browser. Users can access the full functionality of EyeMap through all standard Web browsers, which demonstrates that EyeMap can serve as a useful front-end tool for presenting text corpora and supporting the analysis of large sets of reading data. The current online version is not connected to any database as yet, so users currently cannot change the default data source. However, in the near future, the online system will allow users to upload and select their data source from a set of data corpora. This type of functionality is our ultimate goal and will motivate the design principle of future EyeMap versions.

Implementation and testing

To satisfy the requirement of platform independence, the system was built using several interpreted languages. Interpreted languages give applications some additional flexibility over compiled implementations, such as platform independence, dynamic typing, and smaller executable program size, to list only a few. The data parser, which converts EyeLink EDF data file to a device-independent XML data format, is written in Java, built using JDK (Java Development Kit) 6, and packaged into an executable jar file. EyeMap itself is written in Adobe FLEX and built using Adobe Flash builder 4.0. Adobe Flex is a software development kit (SDK) released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform. Since

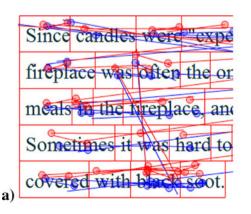
Adobe Flash platform was originally designed to add animation, video, and interactivity to Web pages, FLEX brings many advantages to developers wanting to create user-friendly applications with many different modes of data visualizations, as well as integrated audio and animation, which fully satisfy the design demands of EyeMap. Although maintained by a commercial software company, a free FLEX SDK is available for download, thus making source code written in FLEX freely available.

In addition to manually checking the mapping of fixation location to word and letter locations, we compared the variables created in EyeMap with the analysis output from DataViewer. For this validity check, we randomly picked a subject data set from a current sentence-reading study consisting of 132 trials.

For the EyeMap analysis, the EDF output was converted into XML using EyeMap's edf-asc2xml converter. Word regions were created automatically, on the basis of the information in the text.html file. Fixation and Word Reports were exported, and the resulting CSV files were read into SPSS.

For the DataViewer analysis, EDF data were imported into DataViewer. Since the autosegmentation function in DataViewer did not create word boundaries immediately following the last pixel of the last letter in a word, interest areas had to be defined manually for each trial. Fixation, Saccade, and Interest Area Reports were exported, and resulting Excel files were read into SPSS. Variables from both outputs were compared and showed exact matches. The table in Appendix 2 presents results for some key variables for different word lengths for this one subject.

Fig. 13 Eye scan path before and after the manual drift correction



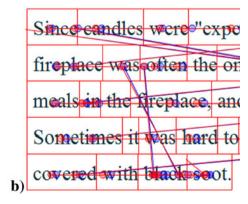
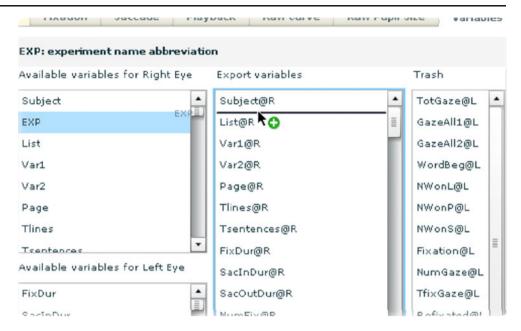




Fig. 14 Creating an export variable list by drag-and-drop in the *export.xml* editor



Related work

Prior to EyeMap, the best-known and most widely used eye movement data analysis tool for reading and related applications was probably the EyeLink DataViewerTM (SR Research Ltd.). The EyeLink family of eye-tracking systems use high-speed cameras with extremely accurate spatial resolution, making them very suitable for reading research. The SR Data Viewer is commercial software, allowing the user to view, filter,

and process EyeLink data. In visualization, it provides several different data-viewing modes such as eye event position and scan path visualization and temporal playback of recording with gaze position overlay. In data analysis, it can generate variables including "interest area dwell time" and fixation progression, first-pass and regressive fixation measures, and so forth. Therefore, DataViewer is a powerful and professional tool for both usability studies and reading research. However, when directly compared, EyeMap offers both comparable

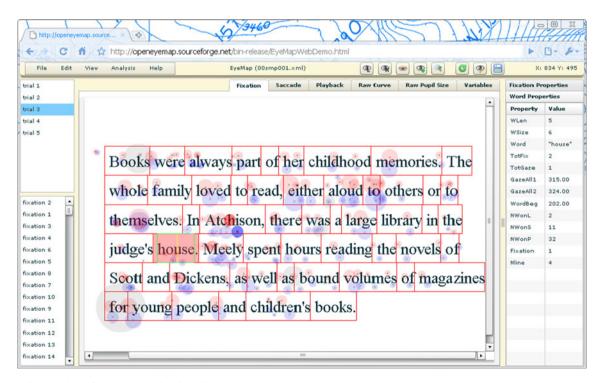


Fig. 15 Online version of EyeMap running in a Chrome browser



features and several distinct additional ones. First, from the visualization point of view, EyeMap is fully compatible with the EyeLink data format and has visualization modes comparable to those provided by DataViewer. Moreover, EyeMap has more precise AoI extraction from text. On the other hand, from an analysis aspect, although DataViewer provides many export variables, most of them are too general for reading psychology or psycholinguistic purposes. In contrast, EyeMap can generate a large number of specific oculomotor measures, representing virtually all common analysis parameters in current basic and applied research (Inhoff & Radach, 1998). Another important difference is that Data Viewer cannot combine binocular data. Left-eye and right-eye data have to be analyzed separately. EyeMap, however, can generate binocular data reports. Data Viewer is also limited to running on Windows and Mac platforms, while EyeMap runs on any currently available platform. It should be noted that EyeMap is not intended to fully replace more comprehensive software solutions such as Data ViewerTM, BeGazeTM, and Studio SoftwareTM, but it can provide a very useful additional tool for the analysis of reading data.

Conclusion

The key innovation in EyeMap involves the development of a freely available analysis and visualization platform specifically focused on eye movement data from reading studies. This focus has allowed for the development of (1) an XML-based open standard for the representation of data from a range of eye-tracking platforms; (2) the implementation of full Unicode support, significantly improving the ease with which different writing systems can be studied; (3) the development of robust techniques for text handling, such as the automatic segmentation of text in both proportional and nonproportional fonts into areas of interest; (4) the development of support for integrating a speech stream with the eye movement record; and (5) a facility for analyzing binocular data in a more integrated way than has heretofore been possible.

While the component technologies of the system are not new, their combination represents a unique and powerful reading analysis platform.

Author Note The development of this software was supported by a John and Pat Hume Postgraduate Scholarship from NUI Maynooth and by grants from the U.S. Department of Education, Institute for Education Sciences, Award R305F100027, Reading for Understanding Research Initiative, and from the German Science Foundation (DFG), HU 292/9-1,2 and GU 1177/1-1.

We thank Ralph Radach and Michael Mayer from the Department of Psychology, Florida State University, and Steson Ho from the Department of Psychology, University of Sydney, for valuable advice and for rigorously testing the software. Thanks are also due Jochen Laubrock and an anonymous reviewer for very helpful comments on an earlier draft.

The software homepage is located at eyemap.tk, while the user manual and installation packages can be retrieved freely from sourceforge.net/projects/openeyemap/files.

Appendix 1

Table 1 EyeMap variables

No.	Variable Name	Level	Description			
1	EXP	Expt	Exptal name abbreviation, taken from 3 rd -5th digit of filename.			
2	List	Expt	Optional, taken from 6th digit of filename.			
3	Subject	Expt	Subject code, taken from first two digits of filename.			
4	Var1	Expt	Optional, taken from 7th digit of filename.			
5	Var2	Expt	Optional, taken from 8th digit of filename.			
6	LineCount_T	Trial	Total number of lines for the trial.			
7	SentenceCount_T	Trial	Total number of sentences for the trial.			
8	TrialNum	Trial	Frial number.			
9	TrialProperty	Trial	Properties defined in the trial.csv file.			
10	WordCount_T	Trial	Total number of words for the trial.			
11	Fixated	Word	1 if the word was fixated, 0 if not.			
12	FixCount_W	Word	Total number of fixations on the word.			
13	GazeCount_W	Word	Total number of gazes (passes) on the word.			
14	LineNum_T	Word	Line number in the current trial.			



Table 1 (continued)

No.	Variable Name	Level	Description				
15	BlinkCount_W	Word	Number of blinks on the word.				
16	SentenceNum_T	Word	Sentence number in the current trial.				
7	TVDur_sac	Word	Total viewing time (TVD), or the sum of all fixations on the word+saccades.				
8	TVDur	Word	Total viewing time (TVD), or the sum of all fixations on the word.				
9	Word	Word	Word, without punctuation.				
0.0	WordBlinkDur	Word	Duration of the blinks in the word.				
1	WordCount_L	Word	Total number of words on the line.				
2	WordCount_S	Word	Total number of words in the sentence.				
3	WordLen_punct	Word	Word Length, in letter, including punctuation.				
4	WordLen	Word	Word length, in letters.				
5	WordLocX	Word	x-pixel location of the word (upper left corner).				
6	WordLocY	Word	y-pixel location of the word (upper left corner).				
7	WordNum_E	Word	Word number for the experiment.				
8	WordNum_L	Word	Word number on the line.				
9	WordNum_S	Word	Word number in the sentence.				
0	WordNum_T	Word	Word number for the trial.				
1	WordProperty	Word	Properties defined in the word.csv file.				
2	FixCount_G	Gaze	Total number of fixation for the current gaze (pass).				
3	GazeBlinkDur	Gaze	Duration of the blinks in the gaze.				
4	GazeDur_sac	Gaze	Gaze duration (GD) of the current gaze (pass), including internal saccades, plus outgoing				
5	GazeDur	Gaze	Gaze duration (GD) of the current gaze (pass).				
6	GazeNum_W	Gaze	Number of the current gaze (Pass) on the word.				
7	GazePupil	Gaze	Mean pupil diameter for the entire gaze.				
8	BlinkCount_G	Gaze	Number of blinks on the gaze.				
9	Refixated	Gaze	1, if FixNum_G				
0	RefixCount	Gaze	Total number of refixations on the word in the current gaze				
1	Blink	Fixation	If 1, then blink before; if 1, then blink after the current fixation.				
2	BlinkDur	Fixation	Duration of the blink.				
3	FixDur	Fixation	Duration of the current fixation.				
4	FixLocX	Fixation	x-pixel average fixation location of the current fixation.				
5	FixLocXBeg	Fixation	x-pixel location of the current fixation at the beginning of that fixation.				
6	FixLocXEnd	Fixation	x-pixel location of the current fixation at the end of that fixation.				
7	FixLocY	Fixation	y-pixel Average fixation location of the current fixation.				
8	FixLocYBeg	Fixation	y-pixel location of the current fixation at the beginning of that fixation.				
9	FixLocYEnd	Fixation	y-pixel location of the current fixation at the end of that fixation.				
0	FixNum_E	Fixation	Fixation number, in the experiment.				
1	FixNum_G	Fixation	Number of the current fixation in the current gaze (pass).				
2	FixNum_S	Fixation	Fixation number in the sentence.				
3	FixNum_T	Fixation	Fixation number for the trial.				
4	FixNum_W	Fixation	Number of the current fixation on the words.				
5	FixPupil	Fixation	Mean pupil diameter for the fixation.				
6	FixStartTime	Fixation	Time stamp of the start of the current fixation (ms).				
7	LandPos_NP	Fixation	(monospaced) Landing position in the word, described in letter units (space before word is				
8	LandPos	Fixation	Landing position in the word, described in letter units (space before word is 0).				
9	LandPosDec_NP	Fixation	(monospaced) Landing position in the word, described in letter units including decimal plac (space before word is 0).				



Table 1 (continued)

No.	Variable Name	Level	Description
60	LandPosDec	Fixation	Landing position in the word, described in letter units including decimal places.
61	LaunchDistBeg_NP	Fixation	(monospaced) Launch distance from the beginning of the current word.
62	LaunchDistBeg	Fixation	Launch distance from the beginning of the current word starting at letter zero, which is the spac in his own mind
63	LaunchDistBegDec_NP	Fixation	(monospaced) Launch distance from the beginning of the current word including decimal places
54	LaunchDistBegDec	Fixation	Launch distance from the beginning of the current word including decimal places.
65	LaunchDistCent_NP	Fixation	(monospaced)Launch distance from the center of the current word.
56	LaunchDistCent	Fixation	Launch distance from the center of the current word.
67	LaunchDistCentDec_NP	Fixation	(monospaced) Launch distance from the center of the current word including decimal places.
68	LaunchDistCentDec	Fixation	Launch distance from the center of the current word including decimal places.
69	LaunchDistEnd_NP	Fixation	(monospaced) Launch distance from the end of the current word.
70	LaunchDistEnd	Fixation	Launch distance from the end of the current word.
71	LaunchDistEndDec_NP	Fixation	(monospaced) Launch distance from the end of the current word including decimal places.
72	LaunchDistEndDec	Fixation	Launch distance from the end of the current word including decimal places.
73	LineSwitch	Fixation	If the current fixation is on a different line than the previous fixation.
74	NxtFixDur	Fixation	Duration of the next fixation.
75	NxtLandPos_NP	Fixation	(monospaced) Landing position of the next fixation in characters.
76	NxtLandPosDec_NP	Fixation	(monospaced) Landing position of the next fixation including decimal places.
77	NxtWordFix	Fixation	Outgoing saccade amplitude in word units, zero if last fixation with the same word.
78	PreFixDur	Fixation	Duration of the previous fixation.
79	PreLandPos_NP	Fixation	(monospaced) Landing position of the previous fixation in characters.
80	PreLandPosDec_NP	Fixation	(monospaced) Landing position of the previous fixation including decimal places.
81	PreWordFix	Fixation	Incoming saccade amplitude in word units, zero if last fixation with the same word.
82	Refixation	Fixation	If 1, then multiple fixations on the word, if 0, only 1 or none).
83	RepairTime	Fixation	Total rereading time on word.
84	ReReading	Fixation	1 if there was a prior fixation on the line to the right of the current word; else, 0.
85	G1F1	Fixation	The first fixation in the 1st gaze on the word.
86	G1Fn	Fixation	The last fixation in the 1st gaze on the word.
87	GnFn	Fixation	The last fixation in the last gaze on the word.
88	SacInAmp	Saccade	Amplitude of the incoming saccade, in letters.
89	SacInAmpX	Saccade	Amplitude of the incoming saccade, <i>x</i> -axis.
90	SacInAmpY	Saccade	Amplitude of the incoming saccade, y-axis.
91	SacInDur	Saccade	Duration of the incoming saccade.
92	SacInInter	Saccade	If 0, then intraword saccade; if 1, then interword saccade.
93	SacInLocBegX	Saccade	Beginning location of the incoming saccade, x-axis.
94	SacInLocBegY	Saccade	Beginning location of the incoming saccade, y-axis.
95	SacInLocEndX	Saccade	End location of the outgoing saccade, x-axis.
96	SacInLocEndY	Saccade	End location of the outgoing saccade, y-axis.
97	SacInNWonP	Saccade	Target NWonP of incoming saccade
98	SacInProg	Saccade	If 1, then incoming saccade is progressive; if 0, then regressive.
99	SacInStartTime	Saccade	Time stamp of the start of the incoming saccade (ms).
100	SacInVel	Saccade	Mean incoming saccade velocity, x-coordinate
101	SacInVel_max	Saccade	Maximum incoming saccade velocity, x-coordinate
102	SacInVelX	Saccade	Average incoming saccade velocity, x-axis.
103	SacInVelY	Saccade	Average incoming saccade velocity, y-axis.
104	SacInWord	Saccade	Word from which the current saccade has exited.



Table 1 (continued)

No.	Variable Name	Level	Description
105	SacInXY	Saccade	Saccade amplitude of the incoming saccade, in Euclidian units.
106	SacOutAmp	Saccade	Amplitude of outgoing saccade
07	SacOutAmpX	Saccade	Amplitude of the outgoing saccade, x-plane.
08	SacOutAmpY	Saccade	Amplitude of the outgoing saccade, y-plane.
09	SacOutDur	Saccade	Duration of the outgoing saccade.
10	SacOutInter	Saccade	If 0, then intraword saccade; if 1, then interword saccade.
11	SacOutLocBegX	Saccade	Beginning location of the outgoing saccade, x-axis.
12	SacOutLocBegY	Saccade	Beginning location of the outgoing saccade, y-axis.
13	SacOutLocEndX	Saccade	End location of the outgoing saccade, <i>x</i> -axis.
14	SacOutLocEndY	Saccade	End location of the outgoing saccade, y-axis.
15	SacOutNWonP	Saccade	Target NWonP of outgoing saccade
16	SacOutProg	Saccade	If 1, then outgoing saccade is progressive; if 0, then regressive.
17	SacOutStartTime	Saccade	Time stamp of the start of the outgoing saccade (ms).
18	SacOutVel	Saccade	Mean outgoing saccade velocity, x-coordinate
19	SacOutVel_max	Saccade	Maximal outgoing saccade velocity, x-coordinate
20	SacOutVelX	Saccade	Mean outgoing saccade velocity, x-coordinate
21	SacOutVelY	Saccade	Mean outgoing saccade velocity, x-coordinate
22	SacOutWord	Saccade	Word to which the current saccade is directed.
23	SacOutXY	Saccade	Saccade amplitude of the outgoing saccade, in Euclidian units.
24	ArticDur	Voice	Articulation duration of the current word.
25	VLauoff10	Voice	Distance from voice offset fixation to the first character of the word that is being spoken.
26	VLauon10	Voice	Distance from voice onset fixation to the first character of the word that is being spoken.
27	VoffFix	Voice	Voice offset fixation.
28	VoiceOffPos_NP	Voice	(monospaced)Letter position at time of the current word's voice offset.
29	VoiceOffPosDec_NP	Voice	(monospaced)Letter position at time of the current word's voice offset, including decimal place
30	VoiceOnPos NP	Voice	(monospaced)Letter position at time of the current word's voice onset.
31	VoiceOnPosDec	Voice	Letter position at time of the current word's voice onset, including decimal places.
32	VoiceOnPosDec NP	Voice	(monospaced)Letter position at time of the current word's voice onset, including decimal place
33	VoiceOffPos	Voice	Letter position at time of the current word's voice offset.
34	VoiceOffPosDec	Voice	Letter position at time of the current word's voice offset, including decimal places.
35	VoiceOnPos	Voice	Letter position of the current word's voice onset.
36	VoiceOffTrialTime	Voice	Trial-based time of the current word's voice offset (ms).
37	VoiceOnTrialTime	Voice	Trial-based time of the current word's voice onset (ms).
38	VoiceOffWord	Voice	Fixated word at the time of the current word's voiced offset.
39	VoiceOnWord	Voice	Fixated word at the time of the current word's voiced onset.
40	VonFix	Voice	Voice onset fixation.
41	FalseStartFix	Voice	False start fixation
42	FalseStartTrialTime	Voice	Trial-based time of the current word's voice false start (ms).
43	FalseStartWord	Voice	Fixated word at the time of the current word's voiced false start.
44	FalseStartPos NP	Voice	(monospaced)Letter position of the current word's voice false start.
45	FalseStartPos	Voice	Letter position of the current word's voice false start.
46	FalseStartPosDec NP	Voice	(monospaced)Letter position at time of the current word's voice false start, including decimal plac
47	FalseStartPosDec	Voice	Letter position of the current word's voice false start, including decimal places.
48	MSGxInc	Message	Increment number of a message.
49	MSGxName	Message	Message name.
50	MSGxStartTime	Message	Time stamp (EDF time) of the message.



Appendix 2

The table below displays a sample of typical reading variables for a randomly selected subject broken down by word length. See Appendix 1 for a definition of the variables

Table 2 Comparison of variable values from SR DataViewer and EyeMap

Mean	EM 2.1800	DV
N 8	2 1900	
Std. Deviation 55.43964 55.440 55.440 55.440 55.440 55.440 000 .000 Minimum 92.00 92 92 92 92 1 1 Maximum 274.00 274 274 274 274 274 1 1 Mean 233.8624 233.86 175.16 175.16 182.79 182.79 1.05 1.05 N 109	2.1000	2.1800
Minimum 92.00 92 92 92 92 92 1 1 Maximum 274.00 274 274 274 274 274 1 1 Mean 233.8624 233.86 175.16 175.16 182.79 182.79 1.05 1.05 N 109	8	8
Maximum 274,00 274 274 274 274 274 274 1 1 2 Mean 233,8624 233,86 175,16 175,16 182,79 182,79 1.05 1.05 N 109	.60444	.60444
2 Mean 233.8624 233.86 175.16 175.16 182.79 182.79 1.05 1.05 N 109 100 100 100 100 100 100 100 1	1.15	1.15
N 109 100 210 210 210 210 210 210 22 22 22 23 23 25 25 35 35 35 35 35 35 35 35 35<	2.95	2.95
Std. Deviation 123.80574 123.806 51.712 51.712 59.337 59.337 .210 .210 Minimum 55.00 55 55 55 55 55 1 1 Maximum 779.00 779 348 348 396 396 2 2 3 Mean 246.8095 246.89 177.77 178.04 183.12 183.38 1.04 1.04 N 357 358 357 358 357 358 357 358 Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 184 184 184 184 184 <tr< td=""><td>2.2926</td><td>2.2926</td></tr<>	2.2926	2.2926
Minimum 55.00 55 55 55 55 55 1 1 Maximum 779.00 779 348 348 396 396 2 2 3 Mean 246.8095 246.89 177.77 178.04 183.12 183.38 1.04 1.04 N 357 358 357 358 357 358 357 358 Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184 184 184 184 184 184 184 184 184 184 184	84	84
Maximum 779.00 779 348 348 396 396 2 2 3 Mean 246.8095 246.89 177.77 178.04 183.12 183.38 1.04 1.04 N 357 358 357 358 357 358 357 358 Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184	2.52116	2.52116
3 Mean 246.8095 246.89 177.77 178.04 183.12 183.38 1.04 1.04 N 357 358 357 358 357 358 357 358 Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184<	.02	.02
N 357 358 357 358 357 358 357 358 Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184 <td>20.11</td> <td>20.11</td>	20.11	20.11
Std. Deviation 134.29925 134.120 62.533 62.661 71.094 71.164 .202 .202 Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184 188	2.3706	2.3790
Minimum 41.00 41 35 35 35 35 1 1 Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184	318	319
Maximum 792.00 792 568 568 648 648 3 3 4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184	1.97661	1.96595
4 Mean 327.0598 327.06 186.34 186.34 219.65 219.65 1.18 1.18 N 184 1	.00	.02
N 184	20.84	20.84
Std. Deviation 183.97019 183.970 66.425 66.425 97.558 97.558 .416 .416 Minimum 36.00 36 36 36 36 1 1 Maximum 1064.00 1064 435 435 660 660 3 3 5 Mean 355.2072 355.21 188.04 188.04 230.80 230.80 1.22 1.22	2.3732	2.3732
Minimum 36.00 36 36 36 36 36 1 1 Maximum 1064.00 1064 435 435 660 660 3 3 5 Mean 355.2072 355.21 188.04 188.04 230.80 230.80 1.22 1.22	170	170
Maximum 1064.00 1064 435 435 660 660 3 3 5 Mean 355.2072 355.21 188.04 188.04 230.80 230.80 1.22 1.22	2.87832	2.87832
5 Mean 355.2072 355.21 188.04 188.04 230.80 230.80 1.22 1.22	.06	.06
	34.34	34.34
N 251 251 251 251 251 251 251 251	2.1608	2.1608
1N 231 231 231 231 231 231 231	242	242
Std. Deviation 198.98327 198.983 73.162 73.162 116.029 116.029 .471 .471	.95467	.95467
Minimum 41.00 41 41 41 41 1 1	.02	.02
Maximum 1238.00 1238 582 582 773 773 4 4	11.02	11.02
6 Mean 381.8077 381.81 202.38 202.38 243.40 243.40 1.27 1.27	2.1655	2.1655
N 208 208 208 208 208 208 208 208 208	195	195
Std. Deviation 232.77356 232.774 66.675 66.675 117.727 117.727 .571 .571	.79746	.79746
Minimum 72.00 72 69 69 69 69 1 1	.01	.01
Maximum 1424.00 1424 576 576 953 953 5 5	5.92	5.92
7 Mean 384.5181 384.52 189.67 189.67 220.17 220.17 1.23 1.23	2.2822	2.2822
N 193 193 193 193 193 193 193 193	188	188
Std. Deviation 261.16047 261.160 63.041 63.041 88.451 88.451 .481 .481	1.34691	1.34691
Minimum 103.00 103 36 36 43 43 1 1	.03	.03
Maximum 2294.00 2294 386 386 697 697 4 4	12.22	12.22
8 Mean 439.4660 439.47 206.14 206.14 268.72 268.72 1.36 1.36	2.3063	2.3063
N 103 103 103 103 103 103 103 103	2.5005	



Table 2 (continued)

R_WordLen		TVDur		FixDurG1F1		GazeDurG1F1		FixCountG1F1		SacInAmpG1F1	
		EM	DV	EM	DV	EM	DV	EM	DV	EM	DV
	Std. Deviation	296.67407	296.674	70.985	70.985	185.240	185.240	.827	.827	.76345	.76345
	Minimum	49.00	49	49	49	49	49	1	1	.08	.08
	Maximum	1705.00	1705	404	404	1203	1203	6	6	5.15	5.15
9	Mean	420.9273	420.93	211.49	211.49	249.13	249.13	1.24	1.24	2.2500	2.2500
	N	55	55	55	55	55	55	55	55	54	54
	Std. Deviation	236.45623	236.456	82.988	82.988	125.783	125.783	.508	.508	.79442	.79442
	Minimum	65.00	65	62	62	65	65	1	1	.28	.28
	Maximum	1006.00	1006	510	510	701	701	3	3	4.63	4.63
10	Mean	535.1212	535.12	197.76	197.76	331.45	331.45	1.73	1.73	2.2447	2.2447
	N	33	33	33	33	33	33	33	33	30	30
	Std. Deviation	308.24268	308.243	84.904	84.904	220.611	220.611	1.039	1.039	1.20611	1.20611
	Minimum	151.00	151	41	41	41	41	1	1	.05	.05
	Maximum	1242.00	1242	445	445	967	967	5	5	7.54	7.54
11	Mean	625.7333	625.73	217.53	217.53	353.60	353.60	1.67	1.67	2.3673	2.3673
	N	15	15	15	15	15	15	15	15	15	15
	Std. Deviation	430.32238	430.322	58.300	58.300	281.706	281.706	.976	.976	1.07526	1.07526
	Minimum	191.00	191	118	118	118	118	1	1	.08	.08
	Maximum	1474.00	1474	378	378	1231	1231	4	4	4.21	4.21
12	Mean	467.7778	467.78	201.33	201.33	291.56	291.56	1.56	1.56	2.2956	2.2956
	N	9	9	9	9	9	9	9	9	9	9
	Std. Deviation	234.58835	234.588	81.185	81.185	177.291	177.291	.726	.726	.79128	.79128
	Minimum	143.00	143	79	79	79	79	1	1	1.52	1.52
	Maximum	910.00	910	357	357	606	606	3	3	3.81	3.81
13	Mean	1209.5000	1209.50	166.50	166.50	867.00	867.00	5.00	5.00		
	N	2	2	2	2	2	2	2	2		
	Std. Deviation	89.80256	89.803	48.790	48.790	4.243	4.243	1.414	1.414		
	Minimum	1146.00	1146	132	132	864	864	4	4		
	Maximum	1273.00	1273	201	201	870	870	6	6		
Total	Mean	340.5468	340.50	189.19	189.25	222.73	222.76	1.20	1.20	2.2769	2.2788
	N	1527	1528	1527	1528	1527	1528	1527	1528	1415	1416
	Std. Deviation	225.34808	225.280	67.405	67.420	118.958	118.927	.522	.522	1.68552	1.68305
	Minimum	36.00	36	35	35	35	35	1	1	.00	.01
	Maximum	2294.00	2294	582	582	1231	1231	6	6	34.34	34.34

References

Bates, R. Istance, H., & Spakov, O. (2005) D2.2 Requirements for the common format of eye movement data. Communication by Gaze Interaction (COGAIN), IST-20030511598: Deliverable 2.2. Available at http://www.cogain.org/results/reports/COGAIN-D2.2.pdf

Duchowski, A. T. (2002). A breadth-first survey of eye-tracking applications. Behavior Research Methods, Instruments, & Computers, 34, 455–470.

Engbert, R., Nuthmann, A., Richter, E., & Kliegl, R. (2005). SWIFT: A dynamical model of saccade generation during reading. *Psychological Review*, 112, 777–813. Halverson, T., & Hornof, A. (2002). VizFix software requirements specification. Eugene: University of Oregon, Computer and Information Science. Retrieved August 3, 2011, from http://www.cs.uoregon.edu/research/cm-hci/VizFix/VizFixSRS.pdf

Hyöna, J., Radach, R., & Deubel, H. (Eds.). (2003). *The mind's eye:*Cognitive and applied aspects of eye movement research.

Oxford: Elsevier Science.

Inhoff, A. W., & Radach, R. (1998). Definition and computation of oculomotor measures in the study of cognitive processes. In G. Underwood (Ed.), Eye guidance in reading and scene perception (pp. 29–54). Oxford: Elsevier.

Johnson, S. C. (1978). YACC: Yet another compiler-compiler. Murray Hill, NJ: Bell Laboratories.



Kennedy, A., & Pynte, J. (2005). Parafoveal-on-foveal effects in normal reading. Vision Research, 45, 153–168.

- Kliegl, R., Nuthmann, A., & Engbert, R. (2006). Tracking the mind during reading: The influence of past, present, and future words on fixation durations. *Journal of Experimental Psychology. General*, 135, 12–35.
- Lehtimaki, T., & Reilly, R. G. (2005). Improving eye movement control in young readers. *Aritifical Intelligence Review, 24,* 477–488.
- Lesk, M. E., & Schmidt, E. B. (1975). Lex: A lexical analyzer generator. Murray Hill, NJ: Bell Laboratories.
- Levine, J., Mason, T., & Brown, D. (1992). Lex & Yacc (2nd ed.). Sebastopol, CA: O'Reilly Media.
- Radach, R., & Kennedy, A. (2004). Theoretical perspectives on eye movements in reading: Past controversies, current issues, and an agenda for the future. *European Journal of Cognitive Psychology*, 16, 3–26.
- Radach, R., Schmitten, C., Glover, L., & Huestegge, L. (2009). How children read for comprehension: Eye movements in developing

- readers. In R. K. Wagner, C. Schatschneider, & C. Phythian-Sence (Eds.), *Beyond decoding: The biological and behavioral foundations of reading comprehension* (pp. 75–106). New York: Guildford Press.
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124, 372–422.
- Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. Psychological Review, 105, 125–157.
- Reilly, R. G., & Radach, R. (2006). Some empirical tests of an interactive activation model of eye movement control in reading. *Journal of Cognitive Systems Research*, 7, 34–55.
- Schattka, K., Radach, R., & Huber, W. (2010). Eye movement correlates of acquired central dyslexia. *Neuropsychologia*, 48, 2959–2973.
- van Gompel, R., Fischer, M., Murray, W., & Hill, R. (Eds.). (2007). Eye movements: A window on mind and brain. Oxford: Elsevier Science.

