Can the E-Z Reader Model Predict Eye Movements Over Code? Towards a Model of Eye Movements Over Source Code

Naser Al Madi
Kent State University
Kent, Ohio, USA
nmadi@kent.edu

Cole S. Peterson
University of Nebraska–Lincoln
Lincoln, Nebraska, USA
cole.scott.peterson@huskers.unl.edu

Bonita Sharif
University of Nebraska–Lincoln
Lincoln, Nebraska, USA
bsharif@unl.edu

Jonathan I. Maletic
Kent State University
Kent, Ohio, USA
jmaletic@kent.edu

ABSTRACT

Studies of eye movements during source code reading have supported the idea that reading source code differs fundamentally from reading natural text. The paper analyzed an existing data set of natural language and source code eye movement data using the E-Z reader model of eye movement control. The results show that the E-Z reader model can be used with natural text and with source code where it provides good predictions of eye movement duration. This result is confirmed by comparing model predictions to eye-movement data from this experiment and calculating the correlation score for each metric. Finally, it was found that gaze duration is influenced by token frequency in code and in natural text. The frequency effect is less pronounced on first fixation duration and single fixation duration. An eye movement control model for source code reading may open the door for tools in education and the industry to enhance program comprehension.

CCS CONCEPTS

• Information systems → Language models; **• Software and its engineering → Software organization and properties**; **• Applied computing → Education**; **• Human-centered computing** → *Human computer interaction (HCI)*.

KEYWORDS

eye-movement model, eye tracking, source code, natural text, empirical study

ACM Reference Format:
Naser Al Madi, Cole S. Peterson, Bonita Sharif, and Jonathan I. Maletic. 2020. Can the E-Z Reader Model Predict Eye Movements Over Code? Towards a Model of Eye Movements Over Source Code. In *EMIP '20: The Seventh International Workshop on Eye Movements in Programming, June 2, 2020, Stuttgart, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3379156.3391983

EMIP '20, June 2, 2020, Stuttgart, Germany

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/3379156.3391983

1 INTRODUCTION

The study of eye movements during reading natural text has enhanced our understanding of the cognitive processes involved in reading. More significantly, the understanding of reading allows for better educational tools for general and special education that enhance the efficiency and productivity of reading comprehension.

The research into the factors and effects that influence eye movements during reading spans over 30 years [Rayner 1998] and led to the development of computational models of eye movement that predict when and where the eyes move across a line of text [Engbert et al. 2005; Reichle et al. 2003]. Such models take in account the role of various cognitive systems such as oculomotor control (eye-movement), lexical access (word identification), and semantic memory to provide an understanding of the interaction between language, eye movement, and cognition [Rayner 1998].

Reading source code shares some aspects with reading natural text, yet the cognitive processes involved in reading source code are much less understood and studied. A contributing factor is the recency of programming languages and the cumulative evidence that shows that source code differs fundamentally in purpose, syntax, semantics, and viewing strategy from natural text [Busjahn et al. 2015, 2014; Liblit et al. 2006; Schulte et al. 2010].

The differences in reading natural text and reading source code are fundamental [Busjahn et al. 2015], prohibiting the automatic extension of natural language research findings and models to programming languages. In this paper we aim to investigate if a prominent eye movement control model that is designed for natural English text (E-Z reader) [Reichle et al. 2003] can predict eye movement on source code. This examination will inform us of the linguistic factors that influence eye movements on source code and if these factors are shared between natural text and source code at the token level. In addition, an eye movement control model for source code reading would open the door for tools in both education and industry to enhance program comprehension. Not only could we use this model to analyze and predict source code readability, but it could also be use it as a baseline or as an "ideal reader" against which to compare novice or expert readers.

We address the following research questions in this paper:

- RQ1: Can the E-Z reader model of eye movement control on natural text predict eye movement over source code?
- RQ2: Is there an influence of token frequency on eye movement in source code?

The results show that the E-Z reader model can be used with natural text and with source code where it provides good predictions of the eye movement in our experiment. This result is confirmed by comparing model predictions to eye-movement data from our experiment and calculating the r-squared correlation score [Freedman et al. 2007] for each eye movement metric. Finally, we find that gaze duration is influenced (with moderate to high correlation) by token frequency in code and in natural text. At the same time, the frequency effect is less pronounced on first fixation duration and single fixation duration.

The contributions from this paper are as follows:

- Demonstrating how a natural text eye movement control model is utilized to predict eye movement over source code.
- Providing evidence that source code on the token level is influenced by token frequency in a similar manner to natural text.

2 BACKGROUND

Computational models of eye movement control provide an account for the interaction between eye movement and cognition. Such models provide an understanding of how linguistic and motor factors influence when and where the eyes move. One of the most notable cognitive models of eye movement control in reading natural text is the E-Z reader model [Reichle et al. 2003]. The E-Z reader model has been used successfully in scene comprehension and other non-reading tasks [Reichle et al. 2012]. In this paper, we evaluate the performance of the E-Z reader in predicting eye movements over source code.

The E-Z reader is built on the assumption that linguistic factors, such as word frequency, predictability, and length, influence eye movement duration and location. The period of time eyes stay stationary on a word/token is called *fixation*, and visual features of text are extracted only during fixations [Reichle et al. 2003]. The frequency effect refers to the difference in reaction times for high-frequency and low-frequency words in which low-frequency words are fixated longer than high-frequency words [Rayner 1998]. This is measured through counting the number of times a word appears in books, articles, and various sources. Word frequency for natural languages has been organized in lexical databases such as the CELEX database [Baayen et al. 1996], and it is measured in words per-million. High-frequency words appear thousands of times in a million words - an example is the word "the" which appears approximately 65,000 times-per-million [Baayen et al. 1996]. An example of a low-frequency word is the word "kin" which appears 3 times in a million words [Baayen et al. 1996]. Word length effect refers to the incremental reading time required as words become longer in the number of characters [Rayner 1998]. Word length effect comes from visual acuity as processing more characters that are distributed further from the center of the fixation takes more time than a shorter word [Reichle et al. 2003].

3 SOURCE CODE FREQUENCY AND LENGTH

In order to investigate if the E-Z reader model of oculomotor control designed for natural text can predict eye movements on source code, we require token frequency, predictability, and token length data for source code. This information can be obtained from lexical databases for natural text, but no such resources exist for source code. Therefore, we used 10 Java repositories that fulfilled the guides of selecting meaningful repositories set by [Munaiah et al. 2017] to determine token frequency in Java.

Table 1: Java repositories and natural text corpus [Project Gutenberg. 2019] used for calculating token/word frequency-in-a-million.

Repository	Files	Lines	Tokens
Ant	1314	304957	1053481
Batik	1651	353516	1185185
Cassandra	2673	586451	2055723
Eclipse	154	25914	77699
Log4J	309	60078	208578
Lucene	8467	1874373	6900196
Maven2	378	60775	206887
Maven3	834	113384	388503
Xalan-J	958	348769	1355646
Xerces2	833	261312	958345
Total	17571	3979251	14390243

Natural Text		Sentences	Words
Gutenberg		98552	2621785

The repositories represent 10 successful Java projects with approximately 3.9 million lines of code resulting in approximately 9.1 million tokens after removing comments from code. For the natural text frequency data we used the Gutenberg corpus [Project Gutenberg. 2019] (retrieved through [Loper and Bird 2002]) with approximately 98,000 sentences resulting in approximately 2.6 million tokens of natural text. Table 1 shows details of the selected Java repositories and natural text corpus.

We estimate the frequency of a word/token in print by counting the number of times the word appeared in the corpus/repositories over the size of the corpus/repositories (normalized by words-in-a-million):

$$\text{frequency(word)} = \frac{count(word)}{count(vocabulary)/1000000}$$

Vocabulary refers to all the words that appear in the corpus or repositories and this count is divided by one million because frequency is calculated in words-per-million. Token length is calculated as a direct count of the number of characters in a word.

4 EXPERIMENT

The central question we seek to answer is whether the E-Z reader model of eye movement control over natural text can predict eye movements on source code. We calculate token frequency and length data for source code and natural text, and run it through the E-Z reader model to generate eye movement predictions that we compare with real data [Busjahn et al. 2015] from our experiment (natural text and source code). The E-Z reader model generates three duration metrics that are important to our discussion here [Reichle et al. 2003]. The duration metrics we use are:

(1) First Fixation Duration (FFD): The duration of the first fixation on a word/token.

(2) Single Fixation Duration (SFD): The duration of the fixation when only one fixation was made on the word/token.
(3) Gaze Duration (GD): The sum of all fixations on word/token N before moving to word/token N+1.

4.1 Experimental Data Set

The data set used in this experiment is from both novice and expert programmers collected in 2015 [Busjahn et al. 2015]. Fourteen novices, who attended a Java beginner course, were recruited for the study. The students' eye movements were recorded at the end of each of the six modules taught, and the stimuli included Java code and English natural text. Each module covered different programming fundamentals and took place over the course of several weeks. Six professionals were recruited from several local companies.

Each participant used a SMI RED-m remote eye tracker that sampled their eye movements at 120 Hz. For each of the novice's sessions, the participants read a set of three programs, two of which were English-like pseudocode and one written in Java. For the expert's session, the participants looked at six programs in total. Two of these programs are the same as the ones viewed by the novices in the latter weeks of the Java's beginner course. The other programs are similar in length to the novice's code samples but are more complex and included more advanced programming constructs.

In order to use the eye tracking data with the E-Z reader model, we need to have token level precision on the stimuli. We map each token of the source code stimuli to an area of interest (AOI). As our source code stimuli is statically displayed for the participants, participant's eye tracking data is easily mapped onto these AOIs and is ready for the E-Z reader model. The first and last word/token are removed as required by the E-Z reader model, and fixations less than 90 ms and greater than 700 ms are also removed [Reichle et al. 2003].

4.2 Duration Results

The duration results are based on examining the ability of the E-Z reader model in predicting eye-movement metrics for natural text and source code. The predictions of the model are based on a simulation with 1000 statistical subjects to determine how well the model can predict observed durations. The 1000 statistical subjects refers to the model pretending to be 1000 different people to generate results that account for some aspects of individual differences in reading.

Starting with natural text, the model is expected to produce good predictions as E-Z reader is intended for natural languages. Figure 1 shows model duration predictions in comparison to observations from our experiment, with mean fixation durations calculated in milliseconds for words in 5 frequency classes. These are the same frequency classes used in eye movement control models like E-Z reader and SWIFT [Engbert et al. 2005; Reichle et al. 2003]. The frequency classes represent words with frequency: 1-9, 10-99, 100-999, 1000-9999, and 10000+. The model appears to resemble observed data, to validate this result we calculate Pearson's correlation coefficient (r^2) between each predicted and observed metric.

- r^2 = 0.57 for First Fixation Duration (FFD) indicating a moderate correlation between observed and predicted values.

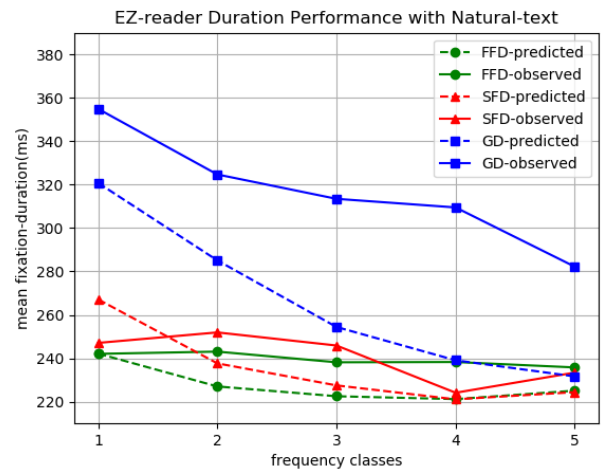

Figure 1: E-Z reader natural text duration predictions in comparison to experiment data. The duration is in milliseconds and the 5 frequency classes correspond to 1-9, 10-99, 100-999, 1000-9999, and 10000+ in words-per-million.

- r^2 = 0.59 for Single Fixation Duration (SFD) indicating a moderate correlation between predicted and observed values.
- r^2 = 0.94 for Gaze Duration (GD) indicating very strong correlation between predicted and observed values.

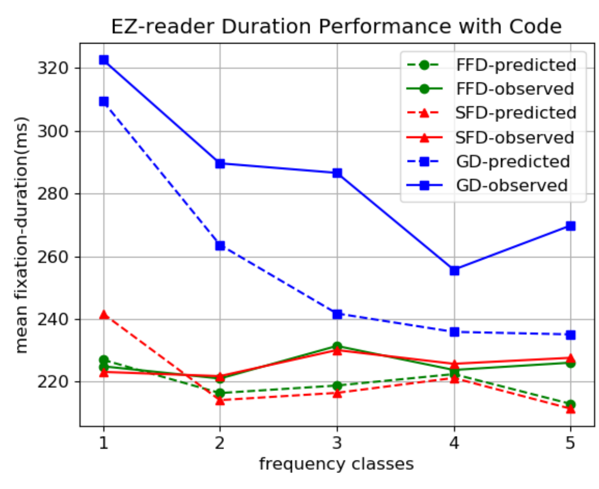

Figure 2: E-Z reader source code duration predictions in comparison to experiment data. The duration is in milliseconds and the 5 frequency classes correspond to 1-9, 10-99, 100-999, 1000-9999, and 10000+ in words-per-million.

Next, we investigated whether the E-Z reader model is able to predict eye movement over source code. The results are presented in Figure 2 which shows the performance of E-Z reader in predicting eye movement duration with source code. Comparing the predicted line and the observed line for each metric we can say that the model captures the observed data. To validate this result, we calculate Pearson's correlation coefficient (r^2) between each predicted and observed metric.

- $r^2 = -0.04$ for First Fixation Duration (FFD) indicating no correlation between observed and predicted values.
- $r^2 = -0.39$ for Single Fixation Duration (SFD) indicating a negative correlation between predicted and observed values.
- $r^2 = 0.92$ for Gaze Duration (GD) indicating very strong correlation between predicted and observed values.

The results for FFD and SFD show no and negative correlation between predictions and the model respectively. This is possibly due to the predictions in the first frequency class. The observed durations are lower than estimated in the first frequency class for both SFD and FFD, nonetheless, GD predictions are highly correlated with the data of our experiment.

5 DISCUSSION

From the results we observe that gaze duration (GD) is higher than model estimations in every case (for both natural text and source code). One possible explanation for this comes from the nature of reading in experiments, as readers tend to spend more time than casual reading. Another explanation could be that reading source code takes more time than natural text as reported in previous studies. [Crosby and Stelovsky 1990] found that subjects needed a significant number of additional fixations to comprehend algorithms in comparison to natural text. In addition, [Busjahn et al. 2011] found that source code received higher fixation durations and more regressions (jumps back) on a significant level statistically.

Almost every eye tracking study with source code indicates that reading code is different from reading natural text. One study that relates most to our question here investigates token length, frequency, and type in source code found that token frequency is not a relevant factor on first fixation duration [Busjahn et al. 2014]. (replicated on the line level in [Peterson et al. 2019]). This finding is supported by our study since FFD and SFD with source code are almost constant over the different frequency classes, but GD is clearly (statistically) affected by token frequency in code and natural text.

Revisiting our research questions, we found evidence to support the use of the E-Z reader model with source code (RQ1) and we found evidence supporting the presence of the frequency effect on gaze duration (GD) in reading source code (RQ2).

6 CONCLUSIONS AND FUTURE WORK

The paper analyzes an existing data set of natural language and source code eye movement data using the E-Z reader model of eye movement control. The results show that the E-Z reader model can be used with source code where it provides good predictions of eye movements in our experiment. This result is confirmed by comparing model predictions to eye movement data from our experiment and calculating the r^2 correlation score for each metric. Finally, we find that gaze duration is influenced by token frequency (with moderate to high correlation) in code and in natural text. The frequency effect is less pronounced on first fixation duration and single fixation duration. Future work consists of an in-depth investigation of frequency, predictability, and length effects in reading source code with novice and experienced programmers.

REFERENCES

R Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. The CELEX lexical database (cd-rom). (1996).

Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye movements in code reading: Relaxing the linear order. In *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE, 255–265.

Teresa Busjahn, Roman Bednarik, and Carsten Schulte. 2014. What influences dwell time during source code reading?: analysis of element type and frequency as factors. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 335–338.

Teresa Busjahn, Carsten Schulte, and Andreas Busjahn. 2011. Analysis of code reading to gain more insight in program comprehension. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*. 1–9.

Martha E Crosby and Jan Stelovsky. 1990. How do we read algorithms? A case study. *Computer* 23, 1 (1990), 25–35.

Ralf Engbert, Antje Nuthmann, Eike M Richter, and Reinhold Kliegl. 2005. SWIFT: a dynamical model of saccade generation during reading. *Psychological review* 112, 4 (2005), 777.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007).

Ben Liblit, Andrew Begel, and Eve Sweetser. 2006. Cognitive Perspectives on the Role of Naming in Computer Programs.. In *PPIG*. Citeseer, 11.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. *arXiv preprint cs/0205028* (2002).

Nuthan Munaiah, Steven Kroh, Craig Cabrey, and Meiyappan Nagappan. 2017. Curating GitHub for engineered software projects. *Empirical Software Engineering* 22, 6 (2017), 3219–3253.

Cole S Peterson, Nahla J Abid, Corey A Bryant, Jonathan I Maletic, and Bonita Sharif. 2019. Factors influencing dwell time during source code reading: a large-scale replication experiment. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM, 38.

Project Gutenberg. 2019. Retrieved September, 2019. www.gutenberg.org

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372.

Erik D Reichle, Alexander Pollatsek, and Keith Rayner. 2012. Using EZ Reader to simulate eye movements in nonreading tasks: A unified framework for understanding the eye–mind link. *Psychological review* 119, 1 (2012), 155.

Erik D Reichle, Keith Rayner, and Alexander Pollatsek. 2003. The EZ Reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and brain sciences* 26, 4 (2003), 445–476.

Carsten Schulte, Tony Clear, Ahmad Taherkhani, Teresa Busjahn, and James H Paterson. 2010. An introduction to program comprehension for computer science educators. In *Proceedings of the 2010 ITiCSE working group reports*. ACM, 65–86.