Eye-tracking Metrics in Software Engineering

Zohreh Sharafi
Génie Informatique et Génie Logiciel
École Polytechnique de Montréal
Montréal, Canada
zohreh.sharafi@polymtl.ca

Timothy Shaffer, Bonita Sharif
Computer Science & Information Systems
Youngstown State University
Ohio, USA
trshaffer@student.ysu.edu, bsharif@ysu.edu

Yann-Gaël Guéhéneuc
Génie Informatique et Génie Logiciel
École Polytechnique de Montréal
Montréal, Canada
yann-gael.gueheneuc@polymtl.ca

***Abstract*—Eye-tracking studies are getting more prevalent in software engineering. Researchers often use different metrics when publishing their results in eye-tracking studies. Even when the same metrics are used, they are given different names, causing difficulties in comparing studies. To encourage replications and facilitate advancing the state of the art, it is important that the metrics used by researchers be clearly and consistently defined in the literature. There is therefore a need for a survey of eye-tracking metrics to support the (future) goal of standardizing eye-tracking metrics. This paper seeks to bring awareness to the use of different metrics along with practical suggestions on using them. It compares and contrasts various eye-tracking metrics used in software engineering. It also provides definitions for common metrics and discusses some metrics that the software engineering community might borrow from other fields.**

I. Introduction

Researchers in software engineering (SE) use eye-tracking technology to study the cognitive processes and efforts involved in different types of SE tasks. An eye tracker (hardware and software) monitors an participant's visual attention via eye-movement data [1], [2]. Eye movements are essential to cognitive processes because they focus the participant's visual attention to the parts of a visual stimulus that are processed by the brain. Visual attention triggers cognitive processes that are required to perform tasks [3]. It is also a proxy for visual effort—a subset of cognitive effort—measured as the amount of visual attention allocated to parts of a visual stimulus. The stimulus in SE studies is shown on a computer screen.

A systematic literature review (SLR) showed that previous eye-tracking studies in SE proposed and used a wide variety of eye-tracking metrics to measure and interpret visual effort required to perform tasks using eye movements [4]. However, the number of *unique* metrics is smaller than it appears because there are no standard names and definitions for many metrics. Often, same metrics have been used in several studies but called by different names. Also, similar names have been used for different metrics. Finally, the lack of an exhaustive list of metrics (with unique names) prevents researchers from appreciating the complexity of eye movements and also causes confusion in choosing the most appropriate metrics for their research. The imprecise names and definitions and the conflicting uses of the metrics make it difficult to compare and/or replicate eye-tracking studies in SE.

For example, notwithstanding the definitions in the following sections, it is confusing that a metric calculating the ratio of the total number of fixations for an area of interest (AOI) in a stimulus (or a set of related AOIs) to the total number of fixations for the whole stimulus is called "Fixation Rate" [5], ON-target ALL-target [6], Ratio of ON-target:All-target Fixation (ROAF) [7], Ratio of fixation count [8], Relevant fixation count [9], and "Time in Region (TIR)" [10].

Consequently, we study exhaustively (to the best of our knowledge) all the ways in which an participant's visual effort has been measured in SE eye-tracking studies and provide unique names and definitions for the used metrics. We also discuss the interpretations of the values of these metrics with references to the literature. We provide practical suggestions on using these metrics and, finally, introduce a list of metrics that SE researchers could borrow from usability studies. Therefore, the contributions of this paper are:

1) Side-by-side comparisons and contrast of existing metrics for visual effort in SE eye-tracking studies.
2) A proposal of new metrics to borrow from other domains, with example applications.
3) A discussion on how to standardize metrics to help compare and replicate eye-tracking studies.

We provide necessary background information on eye tracking in Section II. Section III summarizes previous eye-tracking studies in SE. Section IV presents a list of visual-effort metrics followed by a discussion in Section V. Threats to the validity are reported in Section VI-A and VI-B. Section VII concludes and sketches future studies.

II. Background on Eye Tracking

Eye trackers help assess a participant's visual attention by recording eye movements [1], [2], which show where a participant is looking, the duration, and the sequence in which her attention switches from one location to another. We briefly describe some eye movement terminology.

Fixation: The stabilization of the eye on part of a stimulus for a period of time (200-300 ms). The link between fixations and cognitive processes relies on two assumptions [3]: the immediacy assumption, which states that, as soon as a participant sees the word, she tries to interpret it, and the eye-mind assumption, which states that a participant fixates her attention on the word until she comprehends it.

Saccade: The quick (and continuous) eye movements from one fixation to another. Saccadic eye-movements are extremely rapid (within 40-50 ms). Saccades are usually voluntary. Micro-saccades on the other hand are small jerky eye movements that are involuntary and occur during a long fixation to refresh the participant's visual memory.

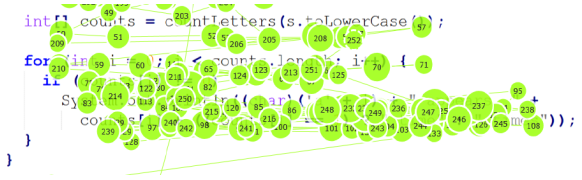

Fig. 1. Scanpath on a code snippet. Fixations are represented by circles, the size of a circle is proportional to the duration of the fixation. Saccades are lines between fixations. Numbers in the circles order the fixations.

Pupil dilation: The widening of the pupil, which allows more light to get into the eye in low light conditions. It also happens when a participant's mood or attitude changes or during complex cognitive tasks [11].

Scanpath: A series of fixations in chronological order that represents an participant's pattern of eye movements.

Researchers study eye-gaze data with respect to certain areas on the stimuli (e.g., diagrams or source code) called Areas of Interest (AOIs). An AOI can be relevant to the correct answer needed from the participant performing a task or can be irrelevant. For example, if we consider a class diagram as a stimulus, an irrelevant AOI can be any class or any notation while a relevant AOI could be the specific class that is relevant to the given task. For source code, it could be any source code element such as method call or identifier. See Figure 1 for an example of a gaze plot overlaid on a code snippet.

III. Previous Eye-Tracking Research in SE

A. Model Stimulus

Several previous eye-tracking studies focused on the comprehension of UML class diagrams with respect to design patterns [6], [7], [12], the impact of layout [13], [14], and the impact of expertise [15]. Petrusel *et al.* [10] focused on the understanding of business process models (BPMN diagrams) while Cagiltay *et al.* [16] performed non-formal inspections of entity relationship diagrams (ERD). In addition, Sharafi *et al.* [17] investigated the efficiency of graphical representations vs. textual ones in modeling and presenting software requirements presented by TROPOS modeling technique.

B. Source code and Text Stimulus

The majority of previous studies focused on the impact of expertise on comprehension and viewing strategies while comparing source-code reading with natural-language text reading [18]–[21]. They reported that source-code reading is different from text reading [8] and that participants have higher fixation time and regression rate when reading source code compared to text [20]. They also reported that novices spend more visual attention on comments than experts while experts spend more time on relevant AOIs [18], [19]. Busjahn *et al.* [22] also performed the first eye-tracking study that adapt several linearity metrics used to analyze natural-language reading to source-code reading. Kevic *et al.* [23] were able to conduct the first eye tracking study on large source code files in an open source system using the Eclipse plugin *iTrace* [24]. *iTrace* automatically maps eye movements to source code elements while maintaining context during scrolling.

Some previous studies analyzed the impact of software visualisation on code comprehension [25] and debugging [26], [27]. They reported that, by helping participants to find relevant AOIs and their dependencies, visualizations guided participants to follow more systematic strategies [25]. Their results also showed that participants with higher performance mainly use graphical representations although they also used text from time to time [26]. Participants with lower performance performed attention switching very frequently [27].

Several previous studies focused on debugging and defect finding tasks [9], [26]–[29]. Turner *et al.* [29] studied the impact of programming languages (C++ vs. Python) and reported that the programming language impacts the visual effort spent by novices compared to experts while working with buggy lines of code. The impact of identifier styles (camel case vs. underscore) on code comprehension has also been investigated [8], [30], [31]. No difference regarding accuracy, time, and effort for comprehension tasks were reported [30], [31] but expertise lessens the impact of identifier styles [30], [31]. Fritz *et al.* [32] combined pupil dilation, blink rates, electrodermal activity, and EEG (electroencephalogram) to characterise and predict task difficulty.

All these previous eye-tracking studies used different types of metrics to compute visual effort for specific tasks. No study summarizes and provides a list of all available metrics along with detailed definitions and suggestions on how to measure visual effort and use the metrics, the main contribution of this paper. In addition, some researchers use the same metric but name it differently. For example, while analyzing the distribution of visual attention on different code elements, all of the previous studies [18]–[21] considered each code element as an AOI and calculated the sums of all fixation durations for each AOI but called this metric differently: fixation time [18]–[20], [26], aggregated fixation time [20], total time of fixations [33], or total dwell time [21]. Conversely, the same name may refer to different metrics. Gaze time either refers to the sum of all fixation durations for an AOI for the whole study [34] or is defined as the sum of all fixation durations [21].

Such ambiguities make comparison and replication of previous studies difficult. This makes it even more important to clarify and standardize the names and definitions of eye-tracking metrics. This paper seeks to bring awareness to these ambiguities and pave the way towards future standardization.

IV. Visual Effort Metrics

When designing an eye-tracking study, researchers in SE must choose adequate metrics to measure the visual effort that is representative of the tasks and stimuli being assessed. We provide a list of visual-effort metrics, their names, and definitions and discuss how previous studies used and interpreted them. We divide metrics into: (1) metrics based on fixations, (2) metrics based on saccades, (3) metrics based on scanpaths, and (4) metrics of pupil size and blink rate.

A. Metrics Based on Fixations

Visual-effort metrics using fixations divide into two groups.

1) Metrics based on the Number of Fixations: Table I presents a list of metrics used in previous studies measuring visual effort based on the number of fixations.

Fixation Count (FC) is the total number of fixations in each AOI. Several studies refer to this metric as the total number of fixations. Goldberg *et al.* [5] reports that a higher number of fixations devoted to a stimulus shows that the search for finding relevant information is not efficient. Previous eye-tracking studies in SE mainly use this metric to find AOIs that attract more visual attention or to report that more visual effort is required to perform a task. When working with text, the fixation count can be adjusted to the text length by dividing the number of fixations by the number of words in the text.

Fixation Rate (FR) [5] is calculated using Equation 1. The Area of Glance (AOG) can be either the whole stimulus, to calculate the ratio of the total number of fixations in one AOI to all fixations, or it can be another AOI, to show the ratio of fixations between two different AOIs. A smaller ratio shows lower efficiency in search tasks: participants spend more effort to find relevant areas [36]. For comprehension tasks, a higher ratio indicates that either the participant shows a great interest in an AOI or that this AOI is difficult to understand [37]. Other names for this metric are "ON-target ALL-target" [6], "Ratio of ON-target:All-target Fixation (ROAF)" [7], "Ratio of Fixation Count" [8], and "Relevant Fixation Count" [9].

$$FR = \frac{\text{Total Number of Fixations in AOI}}{\text{Total Number of Fixations in AOG}} \tag{1}$$

Fixation Spatial Density (SD) [5] is equal to the number of cells containing at least one fixation, divided by the total number of cells, if we consider the stimulus as a grid. It is calculated using Equation 2 where n is the number of cells in the grid and c_i is equal to 1 if the cell number i is visited, otherwise it is 0. It represents the "coverage of an area" and measures the dispersion of the participant's fixations. Smaller spatial density shows less coverage.

$$SD = \frac{\sum_{i=1}^{n} c_i}{n} \tag{2}$$

Convex Hull Area [5] represents the area of the smallest convex set of fixations that contains all a participant's fixations to visualize the spatial distribution of fixations and show the preferred parts of a visual stimulus. It is very sensitive to outliers [4] and even one fixation that deviates from its true location can change the convex hull significantly. Thus, rigourous noise removal is necessary.

2) Metrics based on the Duration of Fixations: The duration of fixations represents the required time to analyze a stimulus [5] and, thus, the depth of processing. Table II presents a list of metrics used by previous studies based on the duration of fixations. It is important to measure both fixation counts and their durations because it is possible to have a low fixation count but a high duration and vice versa.

Average Fixation Duration (AFD) [5] is the sum of the durations of all the fixations divided by the number of fixations, as in Equation 3, where $ET(F_i)$ and $ST(F_i)$ are the end time and start time for a fixation F_i and n is the total number of fixations in a given AOI. Another name for this metric is "Mean Fixation Duration" [20], [38].

$$AFD(AOI) = \frac{\sum_{i=1}^{n}(ET(F_i) - ST(F_i))\text{in AOI}}{n} \tag{3}$$

Ratio of ON-target:All-target Fixation Time (ROAFT) [5] is the sum of the durations of all fixations in an AOI, divided by the total duration of all fixations for the area of glance (AOG), as in Equation 4. An AOG can be the entire stimulus or a set of AOIs. A smaller ratio indicates lower efficiency while searching the stimulus [5]. Other names for this metric are "Proportional Fixation Time (PFT)" [26], "Ratio of Fixation Time" [8], "Relevant Fixation Duration" [9], and "Time in Region (TIR)" [10].

$$ROAFT = \frac{\sum_{i=1}^{n}(ET(F_i) - ST(F_i))\text{in AOI}}{\sum_{j=1}^{n}(ET(F_j) - ST(F_j))\text{in AOG}} \tag{4}$$

Fixation Time (FT), also known as gaze or fixation cluster, is the sum of the durations of all fixations in an AOI. Busjahn *et al.* [21] compute this metric for every AOI visit separately and called it "Dwell Time". They define FT, which they call "Total Dwell Time", as the sum of all dwell times on a AOI over an entire study [21]. Other names for this metric are "Aggregated Fixation Time" [20] and "Total Time of Fixations" [33].

Average Duration of Relevant Fixations (ADRF) is the total duration of the fixations for relevant AOIs as in Equation 5. A corresponding metric exists for non-relevant AOIs and is called "Average Duration of Non-Relevant Fixations (ADNRF)".

$$ADRF = \frac{\text{Fixations Duration of Relevant AOIs}}{\text{Total Number of Relevant AOIs}} \tag{5}$$

Normalised Rate of Relevant Fixations (NRRF) [12] is shown in Equation 6 and allows comparing two or more stimuli with each other. If a stimulus requires more relevant fixations than another, then it requires more visual effort [12]. To adjust for the size of stimulus, this metric must be normalized using the total number of AOIs in the stimulus [12].

$$NRRF = \frac{ADRF}{\left(\frac{\text{Fixation Duration of All AOIs}}{\text{Number of All AOIs}}\right)} \tag{6}$$

B. Metrics Based on Saccades

Table III presents a list of saccade-based metrics used by previous studies to measure visual effort. Higher numbers of saccades indicate more searching [5], [36].

Number of Saccades and *Saccade Duration* are metrics whose definitions are identical to the corresponding fixations-based metrics, see the previous subsection. They also have similar interpretations in relation to the visual effort.

Regressions Rate indicates the percentage of backward saccades of any length [22]. Good readers are characterized by few regressions [22], thus higher regressions rates denote that the participants have difficulty reading and understanding a stimulus [5], [36].

TABLE I. METRICS FOR VISUAL EFFORT BASED ON THE NUMBER OF FIXATIONS

Name	Study	Interpretation	Stimulus
Fixation Count (FC)	(Crosby, 1990) [18]	Higher number indicates beacons (key lines) for comprehension.	Code
	(Crosby, 2002) [19]	Higher number shows more devoted attention to AOI.	Code
	(Uwano, 2006) [28]	Higher number on the whole stimulus while reading and scanning the code leads to find defects faster.	Code
	(Yusuf, 2007) [13]	Higher number indicates poor arrangements of elements in a stimulus which means that more effort is required to explore and navigate.	UML Model
	(Sharif, 2010) [14]	Higher number indicates more visual effort to find defects.	UML Model
	(Sharif, 2012) [9]	Higher number indicates more visual effort to perform the task.	UML Model
	(Sharif, 2013) [8]	Higher number indicates longer processing time to understand source-code phrases.	English text
	(Sharif, 2013) [35]	Higher number indicates more visual effort to perform bug fixing task.	Code
	(Sharafi, 2012) [31]	Higher number indicates more visual effort to recall the name of identifiers.	Code
Fixation Rate (FR)	(Cepeda, 2010) [7]	Higher ratios indicate higher efficiency associated with less effort to find the relevant elements.	UML model
	(Sharif, 2010) [14]	Higher ratio indicates higher efficiency, less effort for the designated layout.	UML model
	(DeSmet, 2012) [6]	Higher ratio indicates higher efficiency, less effort for specific design pattern understanding.	UML Model
	(Sharif, 2012) [9]	Higher ratio indicates less effort.	Code
	(Sharafi, 2012) [31]	Higher ratio shows participant's great interest in analyzing the designated AOIs. While recalling the name of identifiers by answering the multiple choice questions, this metric is used to compare participants based on the ratio of time that they spent analyzing either a correct answer or three wrong choices.	Code
	(Binkley, 2013) [8]	Higher value indicates more effort to understand source-code phrases.	English text
Fixation Spatial Density (SD)	(DeSmet, 2012) [6]	Smaller spatial density indicates more directed search.	Model
	(Soh, 2012) [15]	Smaller value indicates that the participant uses the information gathered from the previous scan or her knowledge, thus she spent less time and effort exploring the diagram.	UML model
Convex hull	(Sharafi, 2012) [31]	Smaller value indicates that the fixations are close from one another thus, less effort is required to find the relevant elements.	Code
	(Soh, 2012) [15]	Smaller value indicates focused fixations, thus less effort is required to find the relevant elements	UML model
	(Sharafi, 2013) [17]	Smaller value indicates less effort to explore the whole model to find the relevant parts of the stimulus.	TROPOS model

TABLE II. METRICS FOR VISUAL EFFORT BASED ON THE DURATION OF FIXATIONS

Name	Study	Interpretation	Stimulus
Average Fixation Duration (AFD)	(Crosby, 1990) [18]	Longer fixations indicates beacons (key lines) for comprehension.	Code
	(Bednarik, 2005) [38]	The distribution of average fixation duration over different areas of interest is different.	Code
	(Cepeda, 2010) [7]	Longer fixations indicate that participants devote more time and effort analyzing and understanding the visual stimulus. Thus, representations that require shorter fixations are more efficient.	UML model
	(Bednarik, 2005) [38]	Longer fixations indicate that more visual effort is required to work with this specific layout.	Code
	(Busjahn, 2011) [20]	Longer fixations indicates a "substantial increase in demands in terms of attentiveness".	Code
	(Soh, 2012) [15]	Longer fixations indicates more overall effort spent by a participant during the task.	UML model
	(Binkley, 2013) [8]	Longer fixations indicates more effort to understand source-code phrases.	English text
	(Cagiltay, 2013) [16]	Longer fixations indicates that the difficulty level of the task is higher.	ERD Model
	(Sharafi, 2013) [17]	Longer fixations indicates more effort to complete the task.	TROPOS model
Ratio of ON-target:All-target Fixation Time (ROAFT)	(Bednarik, 2006) [25]	Higher ratio of source code AOI over visualization AOI indicates the importance of code to participants.	Code
	(Cepeda, 2010) [7]	Higher ratios indicate higher efficiency associated with lower effort to find the relevant elements.	UML model
	(Bednarik, 2012) [26]	The ratio indicates the total amount of time spent on designated area compared to the rest.	Code
	(Sharif, 2012) [9]	The ratio indicates the visual effort to perform the task.	Code
	(Binkley, 2013) [8]	Higher ratio indicates more effort to understand source-code phrases.	English text
	(Petrusel, 2013) [10]	Higher value increases the probability of finding the correct answer.	BPMN Model
Fixation Time (FT)	(Crosby, 1990) [18]	Higher value indicates more relative attention that is devoted to an AOI.	Code
	(Crosby, 2002) [19]	Higher value shows areas that the participant considers important.	Code
	(Uwano, 2006) [28]	Higher value for code reading and scanning leads to find defects faster.	Code
	(Bednarik, 2012) [26]	Higher value indicates more effort.	Code
	(Busjahn, 2014) [21]	Higher value indicates higher attention which denotes rich information and–or higher complexity of the element.	Code
	(Ali, 2015) [33]	Higher value shows areas that the participant considers important.	Code
Average Duration of Relevant Fixations (ADRF)	(Jeanmart, 2009) [12]	Higher value indicates more time (attention) devoted to relevant AOIs.	UML Model
	(DeSmet, 2012) [6]	Higher value indicates more time (attention) devoted to relevant AOIs.	UML Model
Normalised Rate of Relevant Fixations (NRRF)	(Jeanmart, 2009) [12]	Higher rate indicates more effort.	UML Model
	(DeSmet, 2012) [6]	Higher rate indicates more effort.	UML Model
	(Soh, 2012) [15]	Higher rate indicates more effort.	UML model

C. Metrics Based on Scanpaths

Table IV shows scanpath-based metrics for visual effort used by previous studies.

Attention Switching Frequency measures the dynamics of visual attention using the total number of switches between a set of AOIs per minute. A switch happens whenever the participant's focus of attention changes between any AOIs.

Transitional Matrix is a tabular representation of the frequencies of transitions between AOIs [39] computed using Equation 7 in which n is the number of fixations in a AOI (for one cell) and c_i is equal to 1 if the AOI number i is visited and 0 otherwise. To compare two transition matrices, the density of a transition matrix is the number of non-zero matrix cells divided by total number of cells. Higher density indicates extensive search with inefficient scanning on a stimulus, while a sparse matrix is a proxy for an efficient and directed search [5]

$$TM = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j}}{n^2} \tag{7}$$

Edit Distance uses the Levenshtein algorithm, which computes the minimum editing cost to transform one string into another using three basic operations (insertion, deletion, and substitution). (For each operation, the cost of one is considered.) It only uses the location of fixations, not their duration.

TABLE III. METRICS FOR VISUAL EFFORT BASED ON SACCADES.

Name	Study	Interpretation	Stimulus
Number of Saccades	(Fritz, 2014) [32]	It is associated with mental workload and it helps to gain insight into how their eye-movements are influenced by the difficulty of the material.	Code
Saccade Duration	(Fritz, 2014) [32]	It is associated with mental workload and it helps to gain insight into how their eye-movements are influenced by the difficulty of the material.	Code
Regression Rate	(Busjahn, 2011) [20]	Higher regression rate is reported for source code compared to natural language text.	Code
	(Busjahn, 2015) [22]	Regression rate describes non-linear reading	Code

TABLE IV. METRICS FOR VISUAL EFFORT BASED ON SCANPATHS

Name	Study	Interpretation	Stimulus
Attention Switching Frequency	(Bednarik, 2006) [25]	The attention allocation and its switching between AOIs denotes what information and what representation is relevant to the participants during the task.	Code
Transitional Matrix	(DeSmet, 2012) [6]	Frequent transition between AOIs (almost full matrix) indicates inadequate search.	UML Model
Edit Distance	(DeSmet, 2012) [6]	Lower value for edit distance (higher similarity) for a group of participants indicates using the same viewing strategies to scan the visual stimulus.	UML Model
Sequential PAttern Mining (SPAM):	(Sharafi, 2013) [17]	Higher similarity for a group of participants indicates using the same viewing strategies to scan the stimulus.	TROPOS Model
ScanMatch	(Hejmady, 2012) [27]	Higher similarity for a group of participants indicates using the same viewing strategies to scan the stimulus.	Code
	(Busjahn, 2015) [22]	Higher similarity indicates that participants read source code as linearly as natural language text.	Code
Linearity	(Busjahn, 2015) [22]	Expertise impacts to some extent the way participants read source code linearly.	Code

Mathematically, the Levenshtein distance between two strings a and b is calculated recursively as in Equation 8.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if min(i,j) = 0} \\ min = \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_i)} \end{cases} & \text{Otherwise} \end{cases} \quad (8)$$

Sequential PAttern Mining (SPAM) [40] uses the depth-first algorithm to mine and compare scanpaths by considering both fixation locations and durations.

ScanMatch [41] compares scanpaths based on the Needleman-Wunsch algorithm used in bioinformatics to compare sequences of DNA. After adjusting the length of the scanpaths based on the fixations durations using a temporal binning, this metric calculates the similarity score to compare two scanpaths.

Linearity determines a participant's search strategy of a stimulus [36]. For source code, "linearity represents how closely readers follow a text's natural reading order" [22]. It uses eye-movements linearity (left-to-right and top-to bottom) to characterize how developers read source code. A set of local and global metrics exist to measure linearity. For example, local metrics are as follows:

- Vertical Next Text is the percentage of (forward) saccades that happen either on the same line or move only one line below.
- Vertical Later Text is the percentage of (forward) saccades that happen either on the same line or move any number of lines below.
- Horizontal Later Text is the percentage of (forward) saccades that happen on the same line.
- Regression Rate is defined in Section IV-B.
- Line Regression Rate is the percentage of backward saccades that happen on the same line.
- Saccade Length is defined as the average Euclidean distance between consecutive fixations.
- Element Coverage is the percentage of words that a participant puts visual attention on.

Global metrics use the order in which source code is read:

- Story Order is the extent to which the orders of fixations are similar (aligned) to the linear text reading order (left-to-right), using the *ScanMatch* metric [41].
- Execution Order is the extent to which the orders of fixations are similar (aligned) to the program control flow and also uses the *ScanMatch* metric [41].

D. Pupil Size and Blink Rate

Table V presents previous studies that measured visual effort using pupil sizes and blink rates. These two metrics are associated with cognitive workload. Lower blink rates indicate higher workload or attention [32] while higher rates are associated with fatigue [36]. Larger pupil sizes indicate more effort [36]. In addition, Beatty reported that the maximum amplitude of pupil sizes indicates memory and processing load that fluctuates with task difficulty [11].

V. DISCUSSION

A. Data Analysis

Previous eye tracking studies used a combination of the following data analysis approaches to investigate eye-gaze data: (1) start with a hypothesis or a theory and analyze the eye-tracking data to validate it and/or (2) work entirely based on observation without considering any theories in advance [37], [42]. Our review of previous works shows that fixation-based metrics have been mostly used for the first approach, whereby researchers are interested in calculating effort for specific AOIs. Scanpath-based metrics or saccade-based metrics have been mostly used for the second approach, *e.g.,* for evaluating search and navigation strategies. Yet, some existing studies used both approaches, using different sets of metrics on different eye-movement data [15], [17], [31].

Most of the time, defining a set of AOIs is the main step towards analyzing eye-tracking data. Defining an AOI is a subjective task and is based on the researchers' assumptions and experimental goals and conditions. There are no detailed guidelines about defining AOIs [43], especially for SE tasks. Also, the majority of previous studies did not provide details

TABLE V. METRICS FOR VISUAL EFFORT BASED ON THE PUPIL SIZE AND BLINK RATE.

Name	Study	Interpretation	Stimulus
Blink Rate	(Fritz, 2014) [32]	It indicates visual attention. Lower the blink rate shows the higher the mental load or attention.	Code
Pupil Size	(Fritz, 2014) [32]	It indicates cognitive load, memory load, and mental workload. When working with difficult materials, pupil size inclines to increase up to 0.5 mm.	Code

about AOI definition and data extraction. Only, Goldberg *et al.* [43] proposed a set of general guidelines for defining AOIs. Thus, researchers must be careful when defining AOIs and gather and analyze data, especially if the AOIs overlap or are nested. In addition, we encourage researchers in the SE community to explain their choices of AOIs, data extraction, and data analysis approaches in detail. There is also a need for further studies investigating the impact of the type, the granularity, and the analysis of different AOIs for variety of software artifacts and tasks.

B. Metric Popularity

All of the previous studies in SE used fixation-based and scanpath-based metrics. Only a handful of studies used metrics based on saccades because the general consensus in the eye-tracking research community is that cognitive processing and comprehension occur during fixations, while the processing (if any) happening during saccades is very limited. Even for eye-tracking research in usability studies, pupil size and blink rates have been scarcely used because these measures are very sensitive to ambient light level and will be contaminated easily [36]. Only one study used pupil size and blink rate, for measuring task difficulty [32]. It used an EEG device, filtering certain frequencies, to measure blink rates more accurately than using eye-trackers. In addition, it applied noise removal and cleaning techniques on the data. The results were promising and show that the task difficulty can be classified using pupil size and other psycho-physiological factors.

C. Metric Representativeness

We provided a complete list of eye-tracking metrics used in SE research. However, there exists other metrics proposed and used in other domains, such as human-computer interaction (HCI) and usability [5], [36], [37]. These metrics have not been used in SE research yet. We analyze these metrics with respect to their adequacy and (possible) uses in SE research.

We categorize metrics in HCI and usability studies into two main groups [5]: (1) "Measure of processing", which measures the amount of effort required to process, understand, and analyze a stimulus while performing a task and (2) "Measure of search", which measures the amount of effort required to explore and navigate a stimulus when performing a task.

Metrics for the measure of processing are fixation-based metrics, including those presented in Section IV-A. They are mostly and frequently used in SE eye-tracking studies. They provide a single quantitative value to measure the amount of visual effort independently from the type of material that have been used because, to use these metrics, researchers only need to define the AOIs and their relevancy. Thus, these metrics can be used to assess source code or any models, as shown in Tables I and II. Yet, when reviewing the literature on HCI and usability, we found one metric that has not been used in SE.

Saccade-Fixation Ratio (SF Ratio) is a metric for processing that is calculated using Equation 9. It indicates the ratio of searching over processing. A higher value indicates more searching compared to processing. It can be used to compare different layouts (*e.g.,* orthogonal vs. three-cluster vs. multi-cluster layouts for UML class diagrams [14]) or representations (*e.g.,* graphical vs. textual [17]).

$$SFRatio = \frac{\text{Total Saccade Time (Search Time)}}{\text{Total Fixations Time (Processing Time)}} \quad (9)$$

Metrics for search include scanpath-based and saccade-based metrics. Sections IV-B and IV-C present some of these metrics used in SE. There are several other metrics for search in the literature on HCI and usability that have not been used by SE researchers, including:

Average Saccade Amplitude indicates the angular distance that the eye travels and is computed by summing the distances between consecutive fixations, divided by the number of fixations minus one for the stimulus [5]. To filter out micro-saccades, a minimum amplitude is used. This metric can be used to compare two layouts [14] or representations [17] to determine their efficiency for a task. An efficient layout/representation must arrange elements such that participants' scanning to relevant AOIs is minimal [5], [36].

Scanpath Regularity is the ideal situation in which participants follow relevant cues until they reach the desired element. Any deviation from this regular path denotes less efficient search. For SE tasks, a researcher may define a set of relevant AOIs and then consider AOI-based scanpaths instead of ones based on fixations. The most efficient scanpath would be the one that visits all relevant AOIs in a specific order. Irregularity could be measured as either focusing on an irrelevant AOI or following a different path.

Metrics such as *Average Saccade Amplitude* must be used with care because they may not be adequate for SE tasks: they measure the distances between saccades or fixations in numbers of pixels. For SE materials (source code and models), distances between AOIs in numbers of pixels have no meaning because the amplitude is completely dependent on the size of the stimulus and of its elements. For example, two UML class diagrams that differ only by the sizes of the rectangles used to display classes will yield two different values of this metric.

VI. THREATS TO THE VALIDITY

A. Previous Eye-tracking Studies

Several threats limit the validity of previous studies regarding effort measurement based on eye-movement data. We now discuss these potential threats and how previous studies tried to mitigate them. Reported eye-tracking studies use eye-movement data to measure visual effort. One threat to these studies is that there may be other factors that influence the amount of effort put forth by participants to complete their

tasks, including stress (*e.g.,* due to the Hawthorn effect), fatigue, or time of the day. Previous studies tried to mitigate this threat by performing the studies in quiet laboratories. They also avoided long studies to reduce fatigue and limited interactions between participants and researchers.

Like any other human psycho-physiological measure, eye movements may contain gaze data that are unintentional and unconscious. Any distracting events in the environment or in the visual stimulus may impact the eye movements and lead to noise in the collected data and consequently, possibly incorrect interpretations of the data. Previous studies used tasks and stimuli that were not distracting, *i.e.,* without any interactive menus, and performed the experiments in a controlled environment to avoid any distractions.

Another major threat to validity of previous eye-tracking studies is that it is hard to compare results across studies, thereby limiting generalization. The main factor contributing to this lack of generalization was the confusing and redundant naming of visual effort metrics used as dependent variables. To mitigate this threat, we conducted the survey described in this paper. We hope that future studies using eye-trackers will take advantage of this one-stop cross reference for visual effort metrics, thereby eventually standardizing the terms and making it easier to compare results across different studies.

B. This Study

The main threat to validity of this study is whether or not we covered all visual effort metrics that have been proposed or used in previous eye-tracking studies. To reduce the possibility of missing relevant metrics, we performed a SLR [4] by following the process and advice given by Kitchenham [44]. Due to space limitations, we direct the reader to our previous study [4] for the protocol used in the SLR covering 1990–2014, which drives the comparisons done in this study. In this previous study, we did not compare metrics. The first author extracted data from previous eye-tracking studies to provide interpretation of the metrics. To reduce the likelihood of erroneous results, the three other authors checked and validated the extraction and interpretations.

VII. Conclusions and Future Studies

Eye-tracking studies are becoming more prevalent in software engineering. However, a review of the literature [4] shows that previous eye-tracking studies in software engineering proposed and used a wide variety of eye-tracking metrics to measure and interpret visual effort using eye-movement data. Yet, several of these metrics are identical or similar but carry different names. Conversely, some metrics have similar names but different definitions. We surveyed the software engineering literature and, to the best of our knowledge, studied exhaustively all the ways in which an participant's visual effort has been measured in eye-tracking studies. We then present detailed descriptions of the eye-tracking metrics. We also discuss the interpretations of the values of these metrics with references to the literature. We provide some practical suggestions on using these metrics, and finally introduce a list of metrics that software engineering researchers could refer to for HCI and usability studies. Using these lists of existing and potential metrics as well as other suggestions provided, researchers interested in measuring visual effort while performing software engineering related tasks could (1) compare and contrast existing metrics and, therefore, choose the most appropriate one for their studies, (2) borrow "new" metrics from other domains, when appropriate to their studies, and (3) standardize the presentation of the used metrics and their values to help compare and replicate their studies. Hence, we pursue and contribute to the efforts on "standardizing" the reporting of empirical studies [45], [46]. We also aim at reducing the time new researchers would spend to determine what metrics are suitable for their specific study.

A future goal is to move towards a formal standardization procedure for eye-tracking studies in SE. This paper also sets the stage to compare and contrast eye-tracking metrics that will help in future standardization efforts.

Acknowledgment

This study has been partly funded by the Canada Research Chair on Patterns in Mixed-language Systems.

References

[1] A. Duchowski, *Eye tracking methodology: Theory and practice*. Springer-Verlag New York Inc, 2007.

[2] K. Rayner, "Eye movements in reading and information processing: 20 years of research." *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[3] M. A. Just and P. A. Carpenter, "A theory of reading: from eye fixations to comprehension." *Psychological review*, vol. 87, no. 4, p. 329, 1980.

[4] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc, "A systematic literature review on the usage of eye-tracking in software engineering," *Elsevier Journal of Information and Software Technology (IST)*, 2015.

[5] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[6] B. De Smet, L. Lempereur, Z. Sharafi, Y.-G. Guéhéneuc, G. Antoniol, and N. Habra, "Taupe: Visualizing and analyzing eye-tracking data," *Science of Computer Programming*, vol. 79, pp. 260–278, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.scico.2012.01.004

[7] G. Cepeda and Y.-G. Guéhéneuc, "An empirical study on the efficiency of different design pattern representations in uml class diagrams," *Empirical Software Engineering*, vol. 15, no. 5, pp. 493–522, Oct. 2010.

[8] D. Binkley, M. Davis, D. Lawrie, J. I. Maletic, C. Morrell, and B. Sharif, "The impact of identifier style on effort and comprehension," *Empirical Software Engineering*, vol. 18, no. 2, pp. 219–276, Apr. 2013.

[9] B. Sharif, M. Falcone, and J. I. Maletic, "An eye-tracking study on the role of scan time in finding source code defects," in *Proceedings of the Symposium on Eye Tracking Research & Applications*, ser. ETRA '12. New York, NY, USA: ACM, 2012, pp. 381–384.

[10] R. Petrusel and J. Mendling, "Eye-tracking the factors of process model comprehension tasks." in *Proceedings of the Conference on the Advanced Information Systems Engineering*, ser. CAiSE '13. Springer, 2012, pp. 224–239.

[11] J. Beatty, "Task-evoked pupillary responses, processing load, and the structure of processing resources." *Psychological bulletin*, vol. 91, no. 2, p. 276, 1982.

[12] S. Jeanmart, Y.-G. Guéhéneuc, H. A. Sahraoui, and N. Habra, "Impact of the visitor pattern on program comprehension and maintenance." in *Proceedings of 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 69–78.

[13] S. Yusuf, H. H. Kagdi, and J. I. Maletic, "Assessing the comprehension of UML class diagrams via eye tracking." in *Proceeding of 15th IEEE International Conference on Program Comprehension*, ser. ICPC '07. IEEE Computer Society, 2007, pp. 113–122.

[14] B. Sharif and J. I. Maletic, "An eye tracking study on the effects of layout in understanding the role of design patterns." in *Proceedings of the 26th IEEE International Conference on Software Maintenance*. IEEE Computer Society, 2010, pp. 1–10.

[15] Z. Soh, Z. Sharafi, B. V. den Plas, G. C. Porras, Y.-G. Guéhéneuc, and G. Antoniol, "Professional status and expertise for UML class diagram comprehension: An empirical study." in *Proceedings of 20th International Conference on Program Comprehension*, ser. ICPC '13, 2012, pp. 163–172.

[16] N. E. Cagiltay, G. Tokdemir, O. Kilic, and D. Topalli, "Performing and analyzing non-formal inspections of entity relationship diagram (erd)," *Journal of Systems and Software*, vol. 86, no. 8, pp. 2184–2195, Aug. 2013.

[17] Z. Sharafi, A. Marchetto, A. Susi, G. Antoniol, and Y.-G. Guéhéneuc, "An empirical study on the efficiency of graphical vs. textual representations in requirements comprehension." in *Proceeding of 21st International Conference on Program Comprehension*, ser. ICPC '13, 2013, pp. 33–42.

[18] M. E. Crosby and J. Stelovsky, "How do we read algorithms? a case study," *Computer*, vol. 23, no. 1, pp. 24–35, Jan. 1990.

[19] M. E. Crosby, J. Scholtz, and S. Wiedenbeck, "The roles beacons play in comprehension for novice and expert programmers," in *Proceeding of Programmers, 14th Workshop of the Psychology of Programming Interest Group, Brunel University*, 2002, pp. 18–21.

[20] T. Busjahn, C. Schulte, and A. Busjahn, "Analysis of code reading to gain more insight in program comprehension," in *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '11. New York, NY, USA: ACM, 2011, pp. 1–9.

[21] T. Busjahn, R. Bednarik, and C. Schulte, "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 & Applications*, ser. ETRA '14. New York, NY, USA: ACM, 2014, pp. 335–338.

[22] T. Busjahn, R. Bednarik, A. Begel, M. Crosby, J. H. Paterson, C. Schulte, B. Sharif, and S. Tamm, "Eye movements in code reading: Relaxing the linear order," in *Proceedings of 22th International Conference on Program Comprehension*, ser. ICPC '15, 2015.

[23] K. Kevic, B. M. Walters, T. R. Shaffer, B. Sharif, T. Fritz, and D. C. Shepherd, "Tracing software developers eyes and interactions for change tasks," *Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 2015.

[24] T. Shaffer, J. Wise, B. Walters, S. C. Müller, M. Falcone, and B. Sharif, "itrace: Enabling eye tracking on software artifacts within the ide to support software engineering tasks," in *Proceedings of the 10th Joint Meeting (ESEC/FSE 2015)*. ACM, 2015.

[25] R. Bednarik and M. Tukiainen, "An eye-tracking methodology for characterizing program comprehension processes," in *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, ser. ETRA '06. New York, NY, USA: ACM, 2006, pp. 125–132.

[26] R. Bednarik, "Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations," *International Journal of Human-Computer Studies*, vol. 70, no. 2, pp. 143–155, Feb. 2012.

[27] P. Hejmady and N. H. Narayanan, "Visual attention patterns during program debugging with an IDE." in *Proceedings of the 2012 Symposium on Eye Tracking Research & Applications*, ser. ETRA '12. New York, NY, USA: ACM, 2012, pp. 197–200. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168592

[28] H. Uwano, M. Nakamura, A. Monden, and K.-i. Matsumoto, "Analyzing individual performance of source code review using reviewers' eye movement," in *Proceedings of the 2006 symposium on Eye tracking research & applications*, ser. ETRA '06. ACM, 2006, pp. 133–140.

[29] R. Turner, M. Falcone, B. Sharif, and A. Lazar, "An eye-tracking study assessing the comprehension of C++ and Python source code," in *Proceedings of the Symposium on Eye Tracking Research & Applications*, ser. ETRA '14. New York, NY, USA: ACM, 2014, pp. 231–234.

[30] B. Sharif and J. I. Maletic, "An eye tracking study on camelcase and under_score identifier styles." in *Proceeding of 18th IEEE International Conference on Program Comprehension*, ser. ICPC '10. IEEE Computer Society, 2010, pp. 196–205.

[31] Z. Sharafi, Z. Soh, Y.-G. Guéhéneuc, and G. Antoniol, "Women and men - different but equal: On the impact of identifier style on source code reading." in *Proceedings of 20th International Conference on Program Comprehension*, ser. ICPC '13, 2012, pp. 27–36.

[32] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger, "Using psycho-physiological measures to assess task difficulty in software development," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE '14. New York, NY, USA: ACM, 2014, pp. 402–413.

[33] N. Ali, Z. Sharafi, Y.-G. Guéhéneuc, and G. Antoniol, "An empirical study on the importance of source code entities for requirements traceability," *Empirical Software Engineering*, vol. 20, no. 2, pp. 442–478, 2015.

[34] K. Sharma, P. Jermann, M.-A. Nüssli, and P. Dillenbourg, "Understanding collaborative program comprehension: Interlacing gaze and dialogues," in *Proceedings of the 10th Internation Conference on Computer Supported Collaborative Learning*, ser. CSCL '13, 2013.

[35] B. Sharif, G. Jetty, J. Aponte, and E. Parra, "An empirical study assessing the effect of seeit 3D on comprehension." in *Proceeding of 1st IEEE Working Conference on Software Visualization*, ser. VISSOFT '13. IEEE, 2013, pp. 1–10.

[36] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: Current status and future," in *Prospects, Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc*, 2005.

[37] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Mind*, vol. 2, no. 3, p. 4, 2003.

[38] R. Bednarik and M. Tukiainen, "Effects of display blurring on the behavior of novices and experts during program debugging," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '05. New York, NY, USA: ACM, 2005, pp. 1204–1207. [Online]. Available: http://doi.acm.org/10.1145/1056808.1056877

[39] V. Ponsoda, D. Scott, and J. M. Findlay, "A probability vector and transition matrix analysis of eye movements during visual search," *Acta psychologica*, vol. 88, no. 2, pp. 167–185, 1995.

[40] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential pattern mining using a bitmap representation," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 429–435. [Online]. Available: http://doi.acm.org/10.1145/775047.775109

[41] F. Cristino, S. Mathot, J. Theeuwes, and I. D. Gilchrist, "Scanmatch: A novel method for comparing fixation sequences." *Behaviour Research Method*, vol. 42, pp. 692–700, 2010.

[42] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scott, and A. M. Wichansky, "Eye tracking in web search tasks: Design implications," in *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ser. ETRA '02. New York, NY, USA: ACM, 2002, pp. 51–58. [Online]. Available: http://doi.acm.org/10.1145/507072.507082

[43] J. H. Goldberg and J. I. Helfman, "Comparing information graphics: A critical look at eye tracking," in *Proceedings of the 3rd BELIV'10 Workshop: BEyond Time and Errors: Novel evaLuation Methods for Information Visualization*, ser. BELIV '10. New York, NY, USA: ACM, 2010, pp. 71–78. [Online]. Available: http://doi.acm.org/10.1145/2110192.2110203

[44] B. Kitchenham, "Procedures for undertaking systematic reviews," Joint Technical Report, Computer Science Department, Keele University (TR/SE- 0401) and National ICT Australia Ltd, Tech. Rep., 2004.

[45] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting experiments in software engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. Sjøberg, Eds. Springer London, 2008, pp. 201–228.

[46] C. Wohlin, "Writing for synthesis of evidence in empirical software engineering," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM'14. ACM, 2014, pp. 46:1–46:9.