Analyzing and Interpreting Eye Movements in C++

Using Holistic Models of Image Perception

Florian Hauser
florian.hauser@oth-regensburg.de
Technical University of Applied Sciences Regensburg
Regensburg, Germany

Lisa Grabinger
lisa.grabinger@oth-regensburg.de
Technical University of Applied Sciences Regensburg
Regensburg, Germany

Timur Ezer
timur.ezer@oth-regensburg.de
Technical University of Applied Sciences Regensburg
Regensburg, Germany

Jürgen Mottok
juergen.mottok@oth-regensburg.de
Technical University of Applied Sciences Regensburg
Regensburg, Germany

Hans Gruber
hans.gruber@ur.de
University of Regensburg
Regensburg, Germany,University of Turku
Turku, Finland

ABSTRACT

This study uses holistic models of image perception originating from radiology and psychology to analyze and interpret eye movements during code reviews in the C++ programming language. The study design is based on former experiments, but is supplemented by approaches from expertise research. The study utilizes a sample of 34 subjects whose eye movements are recorded by a Tobii Pro Spectrum 600 Hz. The results show that the holistic models of image perception are suitable for application to source code. In addition, it can be observed that the code reviews are conducted in phases, which are characterized by certain strategies (e.g. scan, error detection, ...). Furthermore, experience-related differences can be detected between experts and novices, which emphasize that experts use elaborate strategies and have a comparatively better ability to collect and process information from source code.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; **Empirical studies in visualization**; • **Social and professional topics** → **Computational thinking**; • **Applied computing** → **Education**; • **General and reference** → Surveys and overviews.

KEYWORDS

visual expertise, code reviews, eye tracking, software engineering, holistic models of image perception

ACM Reference Format:

Florian Hauser, Lisa Grabinger, Timur Ezer, Jürgen Mottok, and Hans Gruber. 2024. Analyzing and Interpreting Eye Movements in C++: Using Holistic Models of Image Perception. In *2024 Symposium on Eye Tracking Research and Applications (ETRA '24), June 04–07, 2024, Glasgow, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649902.3655093

ETRA '24, June 04–07, 2024, Glasgow, United Kingdom

ACM ISBN 979-8-4007-0607-3/24/06
https://doi.org/10.1145/3649902.3655093

1 INTRODUCTION

Eye tracking has been utilized in software engineering for over three decades, yielding significant insights into otherwise unobservable cognitive processes [Crosby and Stelovsky 1989; Obaidellah et al. 2018; Sharafi et al. 2015]. However, despite advancements, a challenge persists in its application to source code analysis within software engineering. Unlike engineering sciences in general or other fields like psychology and radiology, which exhibit better standardization and structuring, eye tracking itself lacks consistent definitions and measures, such as fixation and saccade criteria [Duchowski 2017; Holmqvist et al. 2011; Obaidellah et al. 2018; SensoMotoric Instruments 2017; Sharafi et al. 2020, 2015; Tobii Pro 2021]. In computer science and software engineering, this issue also affects data analysis to a certain level [Sharafi et al. 2020].

Psychology and radiology, for example, benefit from established models like the holistic models of image perception, commonly used in interpreting X-ray and MRI studies [Gegenfurtner et al. 2011; Kok 2016; Sheridan and Reingold 2017]. Although initially designed for images, these models are rather adaptable and can be applied in various ways [Sheridan and Reingold 2017].

Thus, this study leverages holistic models of image perception to analyze source code, extending prior research in the field [Hauser et al. 2023]. Unlike its direct predecessor, this study focuses on using C++ programming language instead of C.

2 RELATED WORK

The following section 2 will summarize related work for this study. Initially, eye movements in code reviews will be discussed (see 2.1). Then, holistic models of image perception will be addressed, including the global-focal search model (see 2.2.1), the two-stage detection model (see 2.2.2), and the holistic mode vs. search to find approach (see 2.2.3). Similarities between these models will be identified and explained (see 2.2.4).

2.1 Eye Tracking and Code Reviews

The state of research on eye tracking in code reviews is primarily covered by two meta-studies [Obaidellah et al. 2018; Sharafi et al. 2015]. Together, these cover 44 studies, published between 1989 and

2018. Their key findings can be summarized as follows: Although source code contains letters, numbers, and symbols from natural languages, reading behavior varies significantly between these two visual stimuli [Obaidellah et al. 2018; Sharafi et al. 2015]. While reading a natural text, linear eye movements (in Western languages) from left to right and top to bottom are predominant. However, this method is only found to a limited extent (and mostly among novices) in source code. Experts read more quickly and focus on the source code structure [Begel and Vrzakova 2018; Busjahn et al. 2015, 2014, 2011; Uwano et al. 2006]. Furthermore, it is noted that jumpiness grows with the level of competence [Begel and Vrzakova 2018; Uwano et al. 2006].

Additionally, it is evident from the eye movements of the reviewer that specific reading strategies are employed throughout a code review [Begel and Vrzakova 2018; Hauser et al. 2018, 2020; Nivala et al. 2016; Obaidellah et al. 2018; Sharafi et al. 2015; Sharif et al. 2012; Uwano et al. 2006]. For example, in many circumstances, a so-called *scan* can be noticed at the beginning of a review. This is utilized by the participant to acquire an overview of the source code and grasp its structure [Begel and Vrzakova 2018; Hauser et al. 2018, 2020; Nivala et al. 2016; Sharif et al. 2012; Uwano et al. 2006]. Begel and Vrzakova [Begel and Vrzakova 2018] found recurrent patterns throughout reviews, which they linked to the employment of various strategies. Other studies show that strategies are adapted during a review and depend on the task to be performed, as well as the individual expertise and skills of the reviewer [Bednarik 2012; Bednarik and Tukiainen 2006; Begel and Vrzakova 2018; Busjahn et al. 2014; Hauser et al. 2018, 2020; Nivala et al. 2016; Obaidellah et al. 2018; Peterson et al. 2019; Sharafi et al. 2015; Sharif et al. 2012].

2.2 Holistic Models of Image Perception

To examine visual expertise in code reviews, this study takes into account the findings from previous work [Hauser et al. 2023], the aforementioned literature reviews [Obaidellah et al. 2018; Sharafi et al. 2015] and uses established approaches from radiology and psychology. In this case, the holistic models of image perception are suitable [Sheridan and Reingold 2017].

These models are often used to identify and compare visual strategies of experts and novices [Kok 2016; Reingold and Sheridan 2012; Sheridan and Reingold 2017]. In this paper, three different models are used: the global-focal search model (see Figure 1) [Nodine and Kundel 1987], the two stage-detection model (see Figure 2) [Swensson 1980], and the holistic vs. search-to-find model (see Figure 3) [Kundel et al. 2007].

An essential component of the holistic models of image perception is visual expertise [Gegenfurtner et al. 2011; Kok 2016; Sheridan and Reingold 2017]. This study defines it as a domain-specific competency in an area that necessitates the use of visual methods to complete certain tasks. It is the outcome of extensive training and long-term engagement on a certain subject. During this time, the practitioner adapts to the requirements of the domain and optimizes the relevant cognitive processes of visual information intake and processing in terms of efficiency [Ericsson et al. 1993; Ericsson and Towne 2010; Gegenfurtner et al. 2017, 2011; Gegenfurtner and van Merriënboer 2017; Kok 2016; Reingold and Sheridan 2012; Sheridan and Reingold 2017].

2.2.1 *Global-focal search model.*

The *global-focal search model* was introduced in 1987 [Nodine and Kundel 1987], with subsequent modifications and iterations [Nodine and Mello-Thoms 2000, 2010]. The model focuses on phases of expert visual stimulus processing [Sheridan and Reingold 2017]. In the initial *global* phase, experts conduct a brief scan to gather essential information and compare it with prototypical cases and anomalies (also known as *schemes*). Anomalies are identified and evaluated during this phase. Subsequently, the *focal* phase involves a detailed examination of detected anomalies, marked by observable changes in eye movements [Nodine and Kundel 1987; Sheridan and Reingold 2017]. Nodine and Mello-Thoms state that these visual processes are sequential and may be recursive if necessary before viewers come to a final decision [Nodine and Mello-Thoms 2000, p.869]. Figure 1 is based on former publications and presents a schematic depiction of the *global-focal search model* [Hauser et al. 2023; Nodine and Kundel 1987; Sheridan and Reingold 2017].

2.2.2 *Two-stage detection model.*

The *two-stage detection model* [Swensson 1980] shares similarities with the global-focal search model [Nodine and Mello-Thoms 2000; Nodine and Kundel 1987], both assuming rapid information extraction from visual stimuli using peripheral vision, followed by closer examination using foveal vision [Nodine and Kundel 1987; Reingold and Sheridan 2012; Sheridan and Reingold 2017; Swensson 1980]. While both models involve two phases of visual stimulus processing, Swensson's model does not consider them to be recursive [Nodine and Mello-Thoms 2000; Nodine and Kundel 1987; Swensson 1980]. Instead, the *two-stage detection model* emphasizes prior knowledge as a filtering mechanism influencing eye movements in the initial stage. Vulnerable areas are identified and subjected to detailed analysis in the second phase [Swensson 1980]. Figure 2 illustrates the two-stage detection model [Hauser et al. 2023; Swensson 1980].

2.2.3 *Holistic mode vs. search to find.*

The third relevant model is called *holistic mode vs. search to find* and is depicted in Figure 3 [Hauser et al. 2023; Kundel et al. 2007]. In the *holistic mode*, experts conduct a rapid yet thorough scan of the visual stimulus, enabling them to identify anomalies. Subsequently, areas of interest identified through this method are subjected to more detailed examination by using the *search-to-find* approach. The authors [Kundel et al. 2007] state that the two modes in this model can operate simultaneously. Global information can be processed even when the viewer is already using *search-to-find* to analyze anomalies. The model assumes that the ability to employ the *holistic mode* is correlated with the expertise of the viewers. Novices, lacking certain knowledge and abilities, typically have to use the slower *search-to-find* approach [Kundel et al. 2007; Sheridan and Reingold 2017].

2.2.4 *Common features of holistic models of image perception.*

In their 2017 publication, Sheridan and Reingold [Sheridan and Reingold 2017] describe the flexibility of holistic models of image perception and point out their widespread applicability. They underline that these models can be modified with insights from various disciplines. Additionally, Sheridan and Reingold present the most relevant eye tracking metrics from all three aforementioned holistic

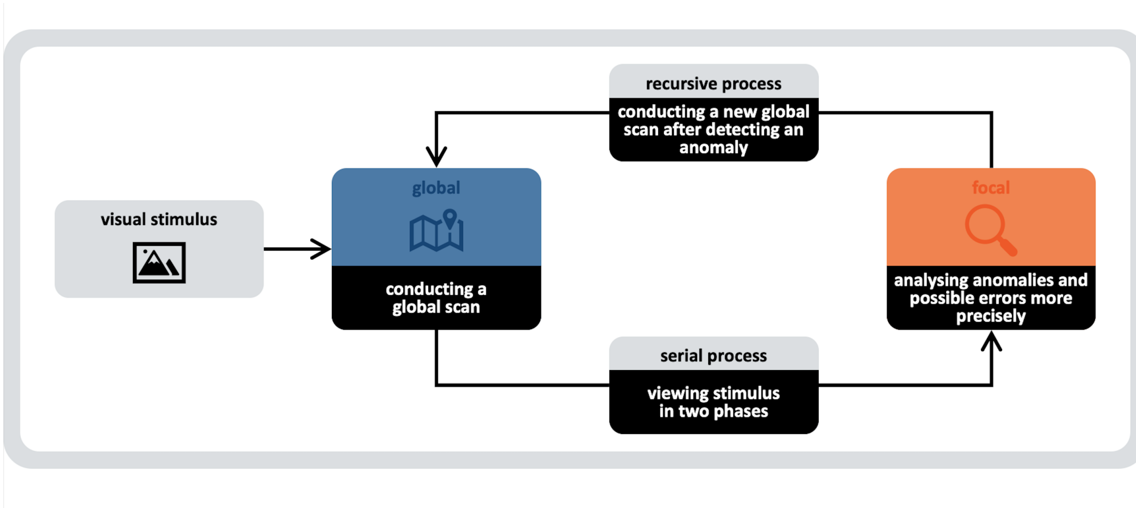

Figure 1: Global-focal search model based on [Nodine and Kundel 1987], taken from [Hauser et al. 2023]

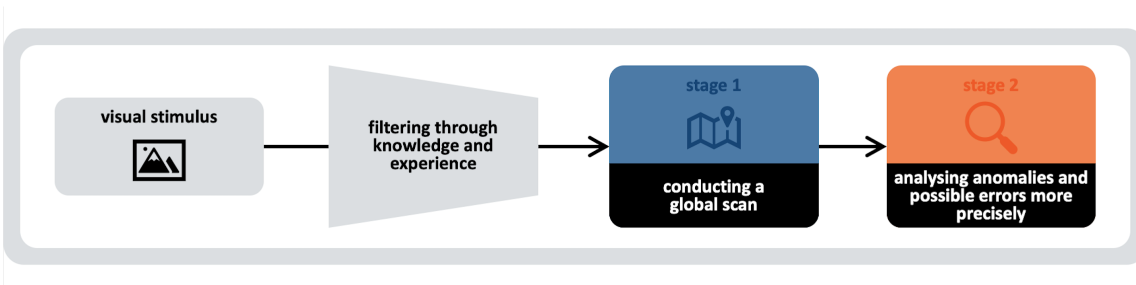

Figure 2: Two-stage detection model following [Swensson 1980], taken from [Hauser et al. 2023]

models and explain their relationship with expertise [Sheridan and Reingold 2017, p.5]:

- *Total viewing times in [ms]:* Experts are expected to gather visual information with less effort and therefore in a shorter period of time.
- *Number of saccades/ fixations:* As expertise increases, the number of fixations and saccades should be reduced.
- *Saccade length:* With more experience, saccades should become longer.
- *Time to first fixation on anomaly in [ms]:* Experts should find errors or anomalies earlier in contrast to novices.
- *Proportional fixation time:* Relevant areas (i.e., anomalies or errors) should be given more attention by experts.
- *Dwell time in [ms]:* As expertise increases, the dwell time for relevant details should become longer.
- *Fixation times in [ms]:* The average fixation duration should decline with more expertise.
- *Fixation rate:* Experts are expected to have a higher number of fixations per seconds as novices.

For the present study, the *number of visits on erroneous lines* is included as an additional metric. Due to limitations of the recording software Tobii Pro Lab (saccade lengths could only be calculated in the used version via a workaround by using additional software [Kanojia 2020]) and changes in the experimental design, the *saccade length*, the *time to first fixation on anomaly* and the *proportional fixation time* are not analyzed in this study.

3 METHODS

This section outlines the methods of the study. It begins with the experimental design 3.1, followed by an overview of the instruments used 3.2, a description of the sample 3.3, and concludes with details of the data collection 3.4.

3.1 Experimental Design

This study partially reproduces the design of former experiments [Hauser et al. 2023, 2018, 2020; Sharif et al. 2012; Uwano et al. 2006]. These studies examined eye movement patterns during a code review. Their design is also used in this study, but slightly modified. In contrast to the publications of Uwano et al. [Uwano et al. 2006] and Sharif et al. [Sharif et al. 2012], this study is examining how the subjects' eye movements during a code review changes. Additionally it puts a stronger focus on the role of experience related differences between expert programmers and novices in computer science. Therefore it uses a contrasting comparison between these

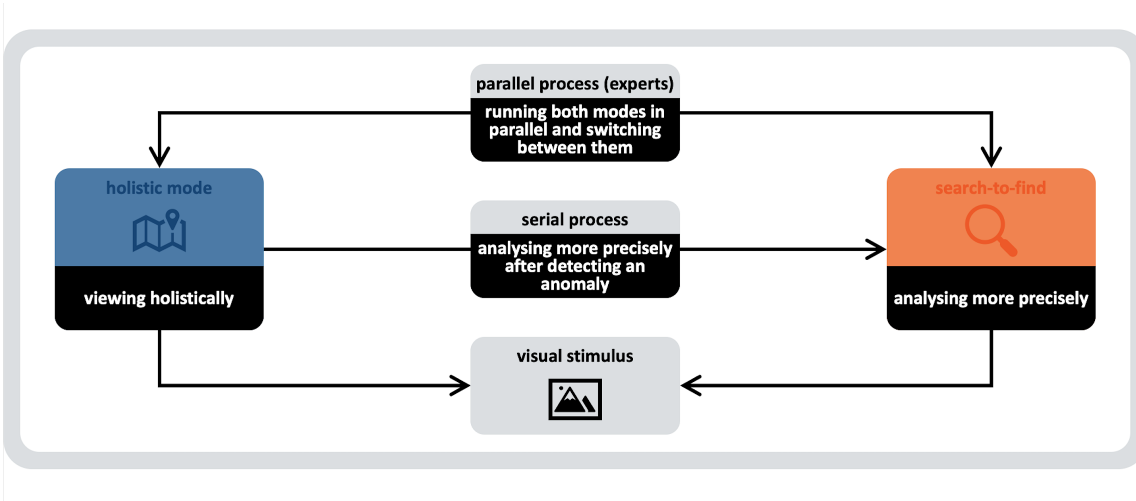

Figure 3: Holistic mode vs. search-to-find following [Kundel et al. 2007], taken from [Hauser et al. 2023]

two groups [Ericsson et al. 1993; Ericsson and Towne 2010; Hauser et al. 2023].

3.2 Instruments

3.2.1 Code examples. A total of eight short code examples were created for the data collection. With regard to the requirements for the examples, it had to be ensured that they were not too long (max. 50 lines of code) due to the scrolling problem with modern eye trackers, so that they could be displayed in full on a normal monitor (23.8"). Furthermore, the examples had to be understandable and solvable for both novices and experts, but also be challenging to a certain extent. In comparison to the predecessor study [Hauser et al. 2023, 2018], correct examples were also created in this case, which on the one hand serve as distractors, but at the same time should also provide insights into the behavior of the test subjects in the absence of an error. As in the previous work, the types of errors included are again logical errors. These are not obvious at first glance and require a deeper understanding of the code. Four code samples contained only one error, while two examples had three errors included. The examples do not use syntax highlighting [Beelders and du Plessis 2016; Peterson et al. 2019; Schorr 2020] and regarding their complexity, they are created in such a way that they can be solved by both novices and experts. Nevertheless, the code review should be challenging to a certain degree for both groups.

3.2.2 Eye tracker. Three Tobii Spectrum with a sampling frequency of 600 Hz are used in this study. In a laboratory and with the best possible calibration, these eye trackers can achieve an accuracy of .100° and maintain it for the duration of an experiment [Tobii Pro 2020]. The three devices are used simultaneously during the study. Every eye tracker is operated by trained personell. After the experiment, the collected data is combined into a common data set.

3.3 Sample

A total of 40 subjects are recruited for this study. After reviewing and cleaning the data sets, six subjects had to be excluded due to calibration problems or data losses. Therefore, data from 34 subjects is still usable. With regard to the age of the test subjects, a range of 17 years is covered, extending from 21 to 38 years. The overall average age is calculated with a mean value of 25.120 (SD = 3.800).

As in the previous study [Hauser et al. 2023, 2018], the subjects' experience is measured on two scales. These were *general programming experience in years* and *professional programming experience in years*. The *general programming experience* (M=5.880; SD = 4.930) refers to the entire period in which the test subjects have been involved in programming. *Professional programming experience* (M=2.250; SD=3.250), on the other hand, refers to the period in which a professional activity is pursued that focuses on programming-specific tasks and with which a significant part of the monthly income is generated.

Based on their *general programming experience*, the sample is divided into two groups. This approach is influenced by expertise research, which uses contrasting comparisons as its primary method [Ericsson et al. 1993; Ericsson and Lehmann 1996; Ericsson and Towne 2010]:

- *Novices (n=18):* To be classified as a novice, subjects had to have less than five years of general programming experience.
- *Experts (n=16):* In order to be classified as expert, the test subjects must have at least five years of general programming experience or more.

The novices were (mostly) undergraduate students, recruited from computer science courses. In contrast, the experts were either professional programmers from local companies or PhD students with a background in engineering and computer science.

3.4 Data Collection

The data collection for this study took place in an eye-tracking classroom. Due to the laboratory setting, the experiment is conducted in a controlled environment and disturbances were reduced to a minimum. In the case of the experts, mobile data collection take place during a C++ user group event. These can be carried out in a suitable location. The data collection is divided into four steps:

(1) *Preparation:* Subjects are briefed and give their consent before receiving instructions for the experiment.
(2) *Questionnaire:* Subjects complete a questionnaire on their programming experience.
(3) *Eye tracking data collection:* Data is collected with Tobii Pro Lab. The experiment is split in four sequences, each with two code stimuli and breaks for the subjects. Calibration is done before each block to ensure best possible accuracy [Holmqvist et al. 2011].
(4) *Stimulus-based interview:* Immediately after data collection, participants view their gaze record and discuss their eye movement and strategies in an interview. Anomalies are noted for further discussion.

The total duration of the individual data collections is depending on individual factors (e.g. *general* and *professional programming experience*) of the test subjects and varies strongly. They cover a time period from around 25 minutes to around 45 minutes. All collected data is anonymized and stored in accordance with the GDPR.

4 ANALYSIS

The data, processed by using Tobii Pro Lab and analyzed in RStudio (version 2023.12.1+402), includes eye tracking and demographic data for each subject (a preliminary analysis of basic eye tracking data is available in a previous publication [Hauser et al. 2020]). In this study, the focus is placed on holistic models of image perception. Therefore the data is divided into three equal thirds based on individual experiment duration and grouped by the subject's experience level. This approach was already used in previous research [Hauser et al. 2023, p.4] and is inspired by Uwano et al.'s findings [Uwano et al. 2006, p.137], indicating that during the first 30% of a code review, 72.8% of the code is typically reviewed. This period, described as "scan," is followed by a more detailed examination focused on error detection. Similar principles are reflected in the holistic models of image perception, which propose that visual stimulus examination unfolds in multiple phases and may involve recursive processes until search results are finally validated [Kundel et al. 2007; Nodine and Kundel 1987; Swensson 1980].

Prior to further analysis of the eye tracking data, the distribution of all relevant metrics is assessed. Kolmogorov-Smirnov and Shapiro-Wilk tests reveal non-normal distribution for all of them. These are also confirmed by examining the corresponding histograms. Consequently, non-parametric Friedman rank sum tests are used to evaluate noticeable differences in metrics across the three phases. The results are presented in Table 1. Significant differences are highlighted in green, while non-significant results are marked in red. Given the presence of outliers, median values are considered to be more reliable then mean values for result interpretation and are included in Table 1.

Additionally, correlations between the subjects' error detection and experience are calculated:

- *Error detection and general programming experience:* r_{BP}=.623, p=.000
- *Error detection and professional programming experience:* r_{BP}=.597, p=.000

Both correlations indicate an experience-related connection with regard to the *error detection*. The more *general* and *professional programming experience* the subjects have, the more likely they will detect errors in the code.

5 CONCLUSION

The following section will discuss the results gained through the use of the holistic models of image perception and how these can be used for the analysis and interpretation of eye movements during a code review (see 5.1). It will also outline what future studies in this area could look like (see 5.2) and address the limitations (see 5.3) of the study presented here.

5.1 Using Holistic Models of Image Perception to Analyze Eye Movements during a Code Review

From the perspective of the holistic models of image perception, the results presented in Table 1 indicate that the code reviews take place in phases and that these differ in terms of the dominant strategy. This is particularly represented by changes in the *fixation rate* and the *number of saccades*.

In case of the fixation rate, it can be observed that it decreases for experts and novices during the experiment. This decline suggests that subjects initially perform a quick scan to get an overview of the code and its structure. After some time, they change to a more detailed examination of anomalies and errors [Kundel et al. 2007; Nodine and Mello-Thoms 2000; Nodine and Kundel 1987; Nodine and Mello-Thoms 2010; Sharif et al. 2012; Swensson 1980; Uwano et al. 2006]. A similar picture emerges when examining the number of saccades: Both experts and novices exhibit a significant increase in the third phase, indicating more thorough code reading. At the begin of the reviews, some (longer) saccades are used to gain an overview, whereas detailed reading, such as error analysis, is associated with an increase in (shorter) saccades [Kundel et al. 2007; Nodine and Mello-Thoms 2000; Nodine and Kundel 1987; Nodine and Mello-Thoms 2010; Sharif et al. 2012; Swensson 1980; Uwano et al. 2006].

Regarding the experience-related differences, the experts and novices use different strategies to carry out the review. These are primarily noticeable in the *average fixation duration, fixation rate, number of saccades, number of visits and dwell time on erroneous lines.* In the case of all metrics, there are indications that the experts use more advanced strategies that enable them to absorb and process information more quickly. However, these results require further in-depth analysis.

In summary, the findings are in line with the previous study [Hauser et al. 2023]: The holistic models of image perception and their metrics offer a solid foundation for the analysis and interpretation of eye movements during a code review. However, refining these models with additional metrics from reading research (e.g. *linearity*, *transitions*) [Busjahn et al. 2015] may be necessary to make them more usable for the requirements of computer science.

Table 1: Results of the Friedman Rank Sum Tests and Medians for Each Eye Tracking Metric or Phase

Eye tracking metric	Group	Differences between phases			Median per phase		
		Phase 1 vs. 2	Phase 1 vs. 3	Phase 2 vs. 3	Phase 1	Phase 2	Phase 3
Number of fixations	Complete	No	No	Yes	191.000	198.500	165.500
	Experts	No	No	No	205.500	226.000	206.000
	Novices	No	Yes	Yes	178.000	170.500	115.000
	Complete: $\chi^2(2)$=16.133, p=.000; Experts: $\chi^2(2)$=6.000, p=.049; Novices: $\chi^2(2)$=13.775, p=.001						
Total fixation duration in [ms]	Complete	No	Yes	Yes	109195.000	108274.000	85681.000
	Experts	No	Yes	Yes	121227.500	126577.000	117492.500
	Novices	No	Yes	Yes	95871.500	93729.500	80500.000
	Complete: $\chi^2(2)$=36.059, p=.000; Experts: $\chi^2(2)$=19.500, p=.000; Novices: $\chi^2(2)$=16.778, p=.000						
Average fixation duration in [ms]	Complete	No	Yes	No	614.157	545.704	521.099
	Experts	No	No	No	584.987	486.226	501.678
	Novices	No	No	No	666.987	569.195	588.188
	Complete: $\chi^2(2)$=9.235, p=.001; Experts: $\chi^2(2)$=3.875, p=.144; Novices: $\chi^2(2)$=5.444, p=.066						
Fixation rate	Complete	Yes	Yes	Yes	1.589	.748	.316
	Experts	Yes	Yes	Yes	1.580	.805	.388
	Novices	Yes	Yes	Yes	1.615	.611	.247
	Complete: $\chi^2(2)$=68.000, p=.000; Experts: $\chi^2(2)$=32.000, p=.000; Novices: $\chi^2(2)$=26.000, p=.000						
Number of saccades	Complete	No	Yes	Yes	170.500	276.000	1341.500
	Experts	No	Yes	Yes	164.000	211.500	911.000
	Novices	No	Yes	Yes	206.500	371.000	1491.000
	Complete: $\chi^2(2)$=32.548, p=.000; Experts: $\chi^2(2)$=13.500, p=.001; Novices: $\chi^2(2)$=20.310, p=.000						
Number of visits on errors	Complete	Yes	No	Yes	4.000	11.500	7.000
	Experts	Yes	No	No	3.500	16.000	7.500
	Novices	No	No	No	4.000	8.000	5.500
	Complete: $\chi^2(2)$=14.970, p=.001; Experts: $\chi^2(2)$=15.129, p=.000; Novices: $\chi^2(2)$=2.771, p=.250						
Dwell time on errors in [ms]	Complete	Yes	No	No	5121.500	6657.500	3702.500
	Experts	Yes	No	No	2997.500	10182.000	3960.000
	Novices	No	No	No	6737.000	4887.000	3438.500
	Complete: $\chi^2(2)$=7.471, p=.024; Experts: $\chi^2(2)$=7.125, p=.028; Novices: $\chi^2(2)$=4.778, p=.092						

5.2 Future Work

Future studies should put their focus on adapting the holistic models of image perception more to the requirements of software engineering or code reviews in specific. For example, this could include a refinement of the metrics provided by former studies [Sheridan and Reingold 2017, p.5] or even the development and addition of new metrics that are specifically created for this purpose [Hauser et al. 2023; Kok 2016; Sharafi et al. 2020, 2015; Sheridan and Reingold 2017]. Considering eye tracking studies on code reviews in general, they do not take the complexity of a code snippet into account [Broy 2006; Kononenko et al. 2016; Obaidellah et al. 2018; Sharafi et al. 2015]. It should also be considered how this can affect the eye movements and how it could be measured via eye tracking.

5.3 Limitations

This work encounters some of the common issues, as do several other eye tracking studies in the field of code reviews [Obaidellah et al. 2018; Sharafi et al. 2015]. 34 subjects make up a small sample size (compared to other domains and other research methods). Furthermore, the code examples are relatively short and synthetically produced for this experiment. It should be considered that programmers usually have to work with much longer codes and have to be able to handle programs with at least 10,000 lines [Broy 2006; Kononenko et al. 2016]. The fact that the data in this study is split into three equal thirds is another limitation. From a statistical point of view, this approach could weaken changes in the subjects viewing strategies. To get more reliable results, the use of certain triggers could be an option for future studies. It should also be taken into consideration to alter the current methodology in order to conduct a more thorough analysis of the time-related variations in the eye tracking metrics. An AI algorithm may be used in this, which could aid in the identification and interpretation of additional phases and patterns.

ACKNOWLEDGMENTS

We thank the funding project FH-Invest (FKZ: 13FH101IN6) run by Prof. Dr. Jürgen Mottok for providing equipment for the eye tracking laboratory and Prof. Dr. Christian Wolff from the University of Regensburg for arranging the laboratory areas. The present paper is based on former results from the EVELIN project (FKZ: 01PL12022F, project sponsor: DLR) which was supported by the Ministry of Education and Research (BMBF) of the Federal Republic of Germany. The in-depth analysis was done in the context of the HASKI project (FKZ: 16DHBKI035), also sponsored by the German Federal Ministry of Education and Research (BMBF).

REFERENCES

Roman Bednarik. 2012. Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations. *International Journal of*

Human Computer Studies 70, 2 (2012), 143–155. https://doi.org/10.1016/j.ijhcs.2011.09.003

Roman Bednarik and Markku Tukiainen. 2006. An eye-tracking methodology for characterizing program comprehension processes. In *Eye Tracking Research and Applications (ETRA)*. ACM, San Diego, 125–132. https://doi.org/10.1145/1117309.1117356

Tanya Beelders and Jean-Pierre du Plessis. 2016. The Influence of Syntax Highlighting on Scanning and Reading Behaviour for Source Code. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT)*, Vol. 26-28-Sept. ACM, Johannesburg, 1–10. https://doi.org/10.1145/2987491.2987536

Andrew Begel and Hana Vrzakova. 2018. Eye movements in code review. In *Proceedings of the Workshop on Eye Movements in Programming (EMIP)*. ACM, Warsaw, 1–5. https://doi.org/10.1145/3216723.3216727

Manfred Broy. 2006. Challenges in automotive software engineering. In *Proceedings of the ACM International Conference on Software Engineering (ICSE)*. ACM, Shanghai, 33–42.

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 *Proceedings of the 23rd IEEE International Conference on Program Comprehension (ICPC)*. IEEE, Florence, 255–265. https://doi.org/10.1109/ICPC.2015.36

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 (ETRA)*. ACM, New York, 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 - Koli Calling '11*. ACM Press, New York, New York, USA, 1. https://doi.org/10.1145/2094131.2094133

Martha E. Crosby and Jan Stelovsky. 1989. The influence of user experience and presentation medium on strategies of viewing algorithms. In *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences.*, Vol. 13. IEEE Comput. Soc. Press, 438–446. https://doi.org/10.1109/HICSS.1989.48025

Andrew T. Duchowski. 2017. *Eye Tracking Methodology* (3rd ed.). Springer, Cham. https://doi.org/10.1007/978-3-319-57883-5 arXiv:arXiv:1011.1669v3

Karl Anders Ericsson, Ralf Thomas Krampe, and Clemens Tesch-Römer. 1993. The role of deliberate practice in the acquisition of expert performance. *Psychological Review* 100, 3 (1993), 363–406. https://graphics8.nytimes.com/images/blogs/freakonomics/pdf/DeliberatePractice(PsychologicalReview).pdf

Karl Anders Ericsson and Andreas C. Lehmann. 1996. Expert and exceptional performance: Evidence of maximal adaptation to task. *Annual Review of Psychology* 47 (1996), 273–305.

Karl Anders Ericsson and Tyler J. Towne. 2010. Expertise. *Wiley Interdisciplinary Reviews: Cognitive Science* 1, 3 (2010), 404–416. https://doi.org/10.1002/wcs.47

Andreas Gegenfurtner, Ellen Kok, Koos van Geel, Anique de Bruin, Halszka Jarodzka, Adam Szulewski, and Jeroen J.G. van Merriënboer. 2017. The challenges of studying visual expertise in medical image diagnosis. *Medical Education* 51, 1 (2017), 97–104. https://doi.org/10.1111/medu.13205

Andreas Gegenfurtner, Erno Lehtinen, and Roger Säljö. 2011. Expertise Differences in the Comprehension of Visualizations: A Meta-Analysis of Eye-Tracking Research in Professional Domains. *Educational Psychology Review* 23, 4 (2011), 523–552. https://doi.org/10.1007/s10648-011-9174-7

Andreas Gegenfurtner and Jeroen J. G. van Merriënboer. 2017. Methodologies for Studying Visual Expertise. *Frontline Learning Research* 5, 3 (2017), 1–13. https://doi.org/10.14786/flr.v5i3.316

Florian Hauser, Lisa Grabinger, Jürgen Mottok, and Hans Gruber. 2023. Visual Expertise in Code Reviews. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, Tübingen, 1–7. https://doi.org/10.1145/3588015.3589189

Florian Hauser, Rebecca Reuter, Ivonne Hutzler, Jürgen Mottok, and Hans Gruber. 2018. Eye Movements in Software Engineering - What Differs the Expert From the Novice?. In *Proceedings of the International Conference of Education, Research and Innovation (ICERI)*. IATED Academy, Seville, 632–642. https://doi.org/10.21125/iceri.2018.1129

Florian Hauser, Stefan Schreistetter, Rebecca Reuter, Jürgen Mottok, Hans Gruber, Kenneth Holmqvist, and Nick Schorr. 2020. Code Reviews in C++. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, Stuttgart, 1–5. https://doi.org/10.1145/3379156.3391980

Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van De Weijer. 2011. *Eye tracking: A comprehensive guide to methods and measures.* Oxford University Press, Oxford.

Deepika Kanojia. 2020. Saccades amplitude or saccade length calculation for each saccade in degree? https://www.researchgate.net/post/Saccades-amplitude-or-saccade-length-calculation-for-each-saccade-in-degree

Ellen M. Kok. 2016. *Developing visual expertise. From Shades of Grey to Diagnostic Reasoning in Radiology.* PhD-Thesis. University of Maastricht, Masstricht.

Oleksii Kononenko, Olga Baysal, and Michael W. Godfrey. 2016. Code review quality. In *Proceedings of the 38th IEEE/ACM International Conference on Software Engineering (ICSE)*. IEEE, Austin, Texas, 1028–1038. https://doi.org/10.1145/2884781.2884840

Harold L. Kundel, Calvin F. Nodine, Emily F. Conant, and Susan P. Weinstein. 2007. Holistic Component of Image Perception in Mammogram Interpretation: Gaze-tracking Study. *Radiology* 242, 2 (feb 2007), 396–402. https://doi.org/10.1148/radiol.2422051997

Markus Nivala, Florian Hauser, Jurgen Mottok, and Hans Gruber. 2016. Developing visual expertise in software engineering: An eye tracking study. In *Proceedings of the Global Engineering Education Conference (EDUCON)*. IEEE, Abu Dhabi, 613–620. https://doi.org/10.1109/EDUCON.2016.7474614

Calvin Nodine and Claudia Mello-Thoms. 2000. The Nature of Expertise in Radiology. In *Handbook of Medical Imaging, Volume 1. Physics and Psychophysics.* SPIE, Bellingham, WA, 859–894. https://doi.org/10.1117/3.832716.ch19

Calvin F. Nodine and Harold L. Kundel. 1987. The cognitive side of visual search in Radiology. In *Eye Movements from Physiology to Cognition.* Elsevier, 573–582. https://doi.org/10.1016/B978-0-444-70113-8.50081-3

Calvin F. Nodine and Claudia Mello-Thoms. 2010. The role of expertise in radiologic image interpretation. In *The Handbook of Medical Image Perception and Techniques*, E Samei and Elizabeth A. Krupinski (Eds.). Cambridge University Press, Cambridge, 139–156.

Unaizah Obaidellah, Mohammed Al Haek, and Peter C.-H. Cheng. 2018. A Survey on the Usage of Eye-Tracking in Computer Programming. *Comput. Surveys* 51, 1 (jan 2018), 1–58. https://doi.org/10.1145/3145904

Cole S. Peterson, Nahla J. Abid, Corey A. Bryant, Jonathan I. Maletic, and Bonita Sharif. 2019. Factors influencing dwell time during source code reading. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research and Applications - ETRA '19*. ACM Press, Denver, CO, 1–4. https://doi.org/10.1145/3314111.3319833

Eyal M. Reingold and Heather Sheridan. 2012. Eye movements and visual expertise in chess and medicine. *The Oxford Handbook of Eye Movements* (2012), 528–550. https://doi.org/10.1093/oxfordhb/9780199539789.013.0029

Nick Schorr. 2020. *Die Rolle von Farbcodierung und Programmierexpertise auf die Fehlersuche im Quellcode Masterarbeit Universität Regensburg.* Ph. D. Dissertation. Universität Regensburg.

SensoMotoric Instruments. 2017. *BeGaze Manual.* Teltow.

Zohreh Sharafi, Bonita Sharif, Yann Gaël Guéhéneuc, Andrew Begel, Roman Bednarik, and Martha Crosby. 2020. A practical guide on conducting eye tracking studies in software engineering. *Empirical Software Engineering* 25, 5 (2020), 3128–3174. https://doi.org/10.1007/s10664-020-09829-4

Zohreh Sharafi, Zéphyrin Soh, and Yann-Gaël Guéhéneuc. 2015. A systematic literature review on the usage of eye-tracking in software engineering. *Information and Software Technology* 67, 7 (2015), 79–107. https://doi.org/10.1016/j.infsof.2015.06.008

Bonita Sharif, Michael Falcone, and Jonathan I Maletic. 2012. An eye-tracking study on the role of scan time in finding source code defects. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*. ACM Press, New York, New York, 381. https://doi.org/10.1145/2168556.2168642

Heather Sheridan and Eyal M. Reingold. 2017. The holistic processing account of visual expertise in medical image perception: A review. *Frontiers in Psychology* 8 (2017), 1–11. https://doi.org/10.3389/fpsyg.2017.01620

Richard G. Swensson. 1980. A two-stage detection model applied to skilled visual search by radiologists. *Perception and Psychophysics* 27, 1 (1980), 11–16. https://doi.org/10.3758/BF03199899

Tobii Pro. 2020. Tobii Spectrum Pro. https://www.tobiipro.com/de/produkte/tobii-pro-spectrum/

Tobii Pro. 2021. Tobii Pro Lab Description. https://www.tobiipro.com/siteassets/tobii-pro/products/software/tobii-pro-lab/Tobii{_}Pro{_}Lab{_}Product{_}Description.pdf/?v=1.181

Hidetake Uwano, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto. 2006. Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)*. ACM, San Diego, CA, 133–140. https://doi.org/http://doi.acm.org/10.1145/1117309.1117357