Visual Attention Tracking During Program Debugging

Roman Bednarik and Markku Tukiainen
Department of Computer Science, University of Joensuu
P.O. Box 111, FI-80101, Joensuu, FINLAND
{roman.bednarik, markku.tukiainen}@cs.joensuu.fi

ABSTRACT
This paper reports on a study which compared two tools for tracking the focus of visual attention - a remote eye tracker and the Restricted Focus Viewer (RFV). The RFV tool blurs the stimuli in order to simulate human vision; the user controls the portion of the screen which is in focus with a computer mouse. Both tools were used by eighteen participants debugging three Java programs for ten minutes each. The results in terms of debugging accuracy and debugging behavior were compared using the restricting view condition of the RFV and a measuring tool as factors.

The results show that while the debugging performance and the distribution of the time spent on areas of interest (AOI) are not influenced by the restricting view condition, the dynamics of programming behavior is different. The number of switches between the AOIs as measured by the RFV significantly differed from those measured by the eye tracker. Also the number of switches under the restricted and unrestricted RFV condition was significantly different. We maintain that the RFV must be used with caution to measure the switches of visual attention.

Author Keywords
Eye-movement tracking, visual attention, psychology of programming.

ACM Classification Keywords
H5.1. Information interfaces and presentation (e.g., HCI): Evaluation/methodology.

INTRODUCTION
The study of visual attention shifts is one of the approaches to understanding the cognitive processes and reasoning. Computer programmers are a typical class of computer users whose mental processing is influenced by the visual inputs their development environments provide. Programmers use debugging tools which provide several representations of a program in adjacent areas of interest. During program comprehension they use these representations to build up a mental model and during debugging they have to coordinate these representations which typically include the source code of a program, some kind of visualization, and the output of the debugger or program execution. Once we can track the visual attention of programmers interacting with the debugger, we can investigate how they coordinate multiple representations, what strategies they adopt, and what their behavioral patterns are. To arrive at the proper conclusions about behavior, it is important to study the limitations and possibilities of the technologies available.

NordiCHI '04, October 23-27, 2004 Tampere, Finland

Related Work
Currently, few tools to track visual attention exist. In recent years, eye-trackers have become common tools for visual attention tracking. Some alternatives to eye trackers, however, have been developed. The Restricted Focus Viewer (RFV) [2] is a visual attention tracking system which displays visual stimuli in a blurred form and allows only a small region to be seen in focus. To get a portion of the stimuli in focus, users have to use a computer mouse to move the focused spot; the focus of visual attention is then thought to be linked to the position of the spot. The RFV tracks the movements of a computer mouse over the stimuli and records them together with the indices of AOIs and timestamps.

In research of tracking the visual attention of programmers, a modified version of RFV was employed in various studies [4, 5, 6]. A Software Development Environment (SDE) was built on top of the RFV and used to track visual attention and investigate the coordination of multiple representations of programmers debugging Java programs. Figure 1 shows a screenshot of the SDE when the restricted condition is on. The code is on the left, the visualization on the top-right, and the output is on the bottom-right. The focused region is displayed over the middle part of the visualization panel.

In our previous study [1], two tools for visual attention tracking were compared: the RFV and the remote eye tracker. Preliminary conclusions indicated that the use of the RFV might affect the behavior of participants, while the debugging performance seemed to be unaffected. Further, our results indicated that for the visual attention tracking in multiple-representation displays, the RFV might not accurately measure the data as expected.

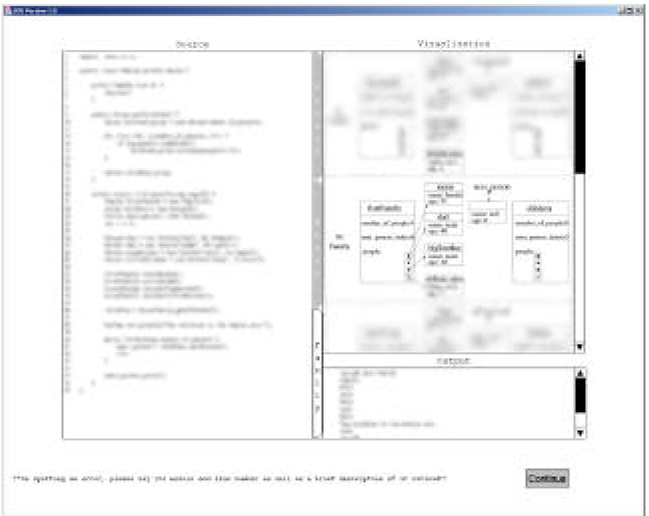

Figure 1. Software debugging environment using RFV.

The aim of the present study was to closely compare two tools for measuring visual attention, the RFV and the eye tracker. Besides, we aimed to further verify the results obtained in a previous study [1], using more participants. Our working hypotheses were: *using the restricted focus view, the RFV-based tool can change the strategies and behavior of a programmer* and *that the RFV does not accurately measures all visual attention switches.*

METHOD

We analyzed and compared the locations of the focus of visual attention as measured by the eye tracker and by the RFV. The RFV-tool links the visual attention focus to the position of a mouse-controlled focused spot in the otherwise blurred stimuli; the eye tracker reports the point of gaze, thought to be linked to the focus of visual attention.

Design and Participants

A totally-within subject design was used with two factors (*RFV restricting condition* and *measuring tool*) and four dependent variables (*errors spotted*, *accumulated fixation time*, *mean fixation duration*, and *switching frequency as measured by RFV and by eye tracker*). The accumulated fixation time is the total time spent during a session the participant is fixating an AOI. For an AOI, all fixations are summed and the number is divided by the total fixation count throughout the experiment, giving the mean fixation duration. The switching frequency refers to the average number of switches per minute between each of the AOIs, as measured by a tracking tool. Most of the results were analyzed by performing ANOVA or paired samples t-tests.

In the study a total of 18 participants were recruited from population of students, researchers, and teachers from the authors' department. All subjects had normal or corrected-to-normal vision and never had taken part in an eye tracking experiment. The average age was 25.3 (SD=4.4) years. Three of participants were females. The programming and Java experience varied from just passing a Java course and having little experience to professionals working in programming related careers. The average programming experience in months was 78.7 (SD=34.7), and the average Java experience was 11.5 (SD=14.8) months.

Procedure

Before the experiment, participants had to pass an automatic eye-tracking calibration procedure. After that, the participants read detailed instruction about the experiment and the environment used. Three programs were debugged. The first warm-up session was performed under the RFV restricted view condition (RFV-on) so that the participant could become familiar with controlling the focused spot and operating the debugging environment. Then, the two main debugging sessions were performed; one session was performed under the RFV-on condition, the other session was performed under the RFV-off condition. The order of the programs and conditions was counterbalanced.

Each session had two phases. First, the specification of the program was displayed which described the problem the program was supposed to solve and the approach to the solution. Two sample interactions were provided - the desired behavior and actual behavior of the program. Second, the participants were given ten minutes to debug the program and were instructed to find as many errors as possible and to report them aloud.

Materials and Setup

The target programs were identical to those used in [4]. The object of the warm-up program was to determine whether a point was inside a rectangle. The first program printed out the names of the children of a sample family and the second program counted the cash in a register till which gave subtotals for the different denominations. In the previous study [4], two versions of the target programs and several visualizations were used. In our experiment, we used the less sophisticated versions of the programs and graphical functional representations. The two main target programs were seeded with four errors each; the warm-up program contained two errors. The programs contained no syntactical error and participants were notified of this.

For the eye tracking, the remote Tobii ET-1750 (sampling at 30Hz) eye tracker was used. The eye tracking data were collected throughout the whole experiment; the RFV collected data only in the RFV-on condition. The AOIs were defined to correspond with the three main panels in the SDE window: the code, visualization, and the output panel.

The software debugging environment (SDE) used in the previous studies [4, 5, 6] was employed for the experiment as a source of stimuli. In these studies and in the present experiment, the program code, the visualization, and output were pre-computed and static.

RESULTS

Debugging performance

The results in terms of debugging performance show that the total number of errors spotted under the RFV-on condition was 46 (mean=2.56, SD=1.10) and under the RFV-off condition was 44 (mean=2.44, SD=1.04) out of maximum 72 errors. There was no significant difference in the average number of errors found between the conditions of restricted and unrestricted view (t(17) = 0.44, *ns*), the grand mean was 2.5 (SD=1.06).

Debugging behavior

The debugging behavior was measured by the eye tracker under the RFV-on/RFV-off and by the RFV-tool under the RFV-on condition. Figure 2 presents the proportion of accumulated time spent on fixating the AOIs as measured by the eye tracker. The effect of the RFV condition on the proportional times spent on the areas of interest was not significant: there was about the same distribution of proportional times regardless of whether the stimuli were blurred or not.

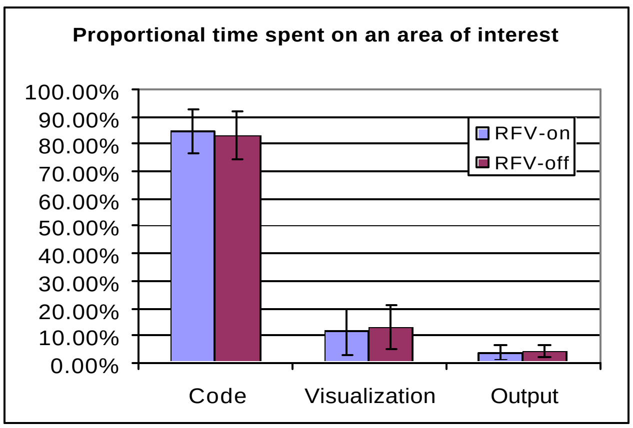

Figure 2. Proportion of time spent on the areas of interest

Figure 3 presents the behavior in terms of number of switches between the AOIs per minute. The results under the RFV-on condition were measured by the RFV-tool. The results under both conditions were measured by the eye tracker. The term "switch" refers to the change of focus between the AOIs, here between the code, visualization, and output panels. Single factor ANOVA was run for the average number of switches. The average number of switches measured significantly differed (F(2,17)=18.37, *p*<.001). Comparing the average number of switches as reported by the measuring tool used under the restricted view condition (RFV-on), there was a significant difference (t(17) = 5.51, *p*<.001) between the RFV and eye tracker. The difference of the number of average switches per minute regarding the condition (RFV-on/RFV-off) was also significant, t(17) =2.42, *p*<.001.

To further study the differences, we analyzed the switching behavior in terms of switches per minute between each of the AOIs. Three two-way ANOVAs revealed a significant effect of the measurement and RFV condition to the number of switches between each AOI (F(2,5)=38.2, *p*<.001), a significant effect of the tool used under the RFV-on condition (F(1,5)=50.7, *p*<.001), a significant interaction between the tool used and the number of switches between each of the AOIs under the RFV-on condition (F(1,5)=5.3, *p*<.001), and a significant effect of the restricting view condition to the number of switches between each of the AOIs as measured by the eye tracker (F(1,5)=9.3, *p*<.01). The numbers of switches per minute between each of the AOIs under as measured by the RFV-tool and by the eye tracker revealed were not correlated.

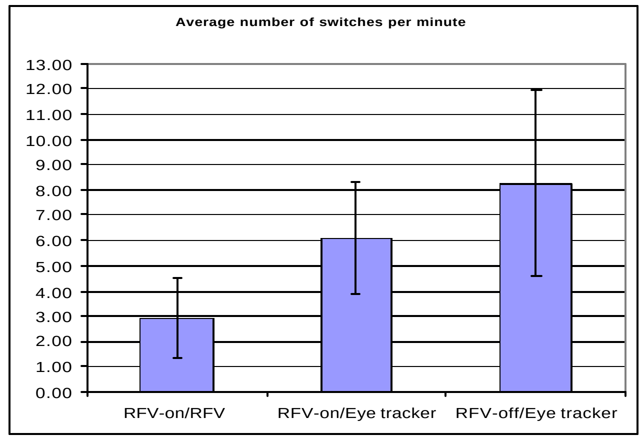

Figure 3. Average number of switches as measured by the RFV under RFV-on, and as measured by the eye tracker under RFV-on and RFV-off

The mean fixation durations measured by the eye tracker are shown in Figure 4. They were obtained by dividing the accumulative duration by the fixation count for each AOI. The two-way ANOVA revealed a significant difference in the mean fixation duration between the individual AOIs (F(1,2)=6.2, *p*<.01); however, the effect of the condition was not significant (F(1,2)=1.9, *ns*). All mean fixation durations under the RFV-off condition were lower than under the RFV-on condition, the effect of RFV condition was nearly significant for the output panel (t(17)= 1.62, *p*<ns) and significant for the overall mean fixation duration (t(17) = 2.09, *p*<.01). The overall mean fixation durations were also significantly correlated (r(18) = 0.857, *p*<.01) between the conditions.

DISCUSSION

The purpose of this experiment was to compare two tools for tracking visual attention measured by the eye tracker and by the RFV tool. The RFV relates the focus of visual attention to the location of a fully focused area within the blurred stimuli images; the eye tracker reports the coordinates of point of gaze which is thought to be connected to the focus of visual attention.

The results show that the error-finding performance and distribution of the fixation times is not influenced by presenting the stimuli in a blurred form. The distributions of the times spent over AOIs as reported by the eye tracker show the consistent pattern of behavior and confirm the results obtained in other studies.

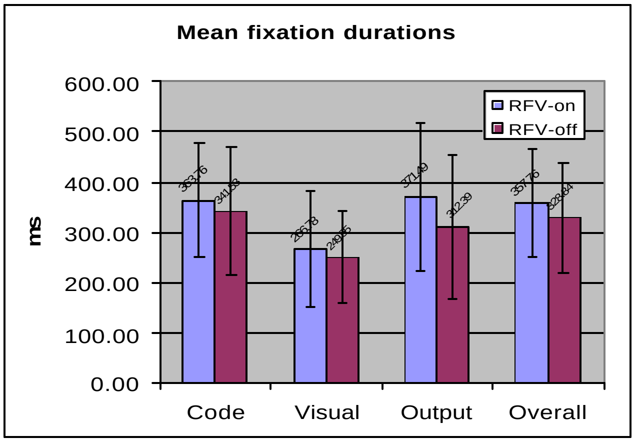

Figure 4. Mean fixation duration for each area of interest

The dynamics of the switching behavior as reported by the RFV tool and by the eye tracker differ significantly. The eye movement protocols may indicate where these differences stem from. Our explanation is that the participants often briefly look onto the blurred areas without moving the focused spot. Therefore, the RFV can not register these changes which most probably serve as a refresher of the mental images of the stimuli. A typical situation of the problem is shown in Figure 5. The participant has changed the visual focus and is fixating on the blurred right-bottom output panel, as indicated by two saccades and following fixation, while the focus as measured by the RFV is in the middle of the code on the left. The second difference is clear when comparing the number of switches with and without the restricted focus view. Participants changed the focus of visual attention more often when the stimuli were presented in focus.

The distribution of mean fixation duration shows a decrease in the duration while the stimuli are unrestricted and a significant decrease of fixation duration over the AIO containing the visualization. In some studies, e.g. [3], the fixation duration mean is thought to be related to the participants' difficulty with extracting the information. This result needs to be analyzed in further studies.

CONCLUSION

We compared two tools for tracking visual attention, the Restricted Focus Viewer and the remote eye tracker, to investigate their possibilities and limitations in the context of software debugging environment. We investigated the data provided by these tools and analyzed the changes in the behavior of participants when the stimuli are presented in a blurred form.

The results indicate that blurring of the stimuli does not influence the error-finding performance of participants and the distribution of times spent on the areas of interest. However, the results show that there are changes in the dynamics of the switching behavior and mean fixation duration. Under the restricted view condition the RFV-tool reports different dynamics in the attention switching than that is measured by the eye tracker

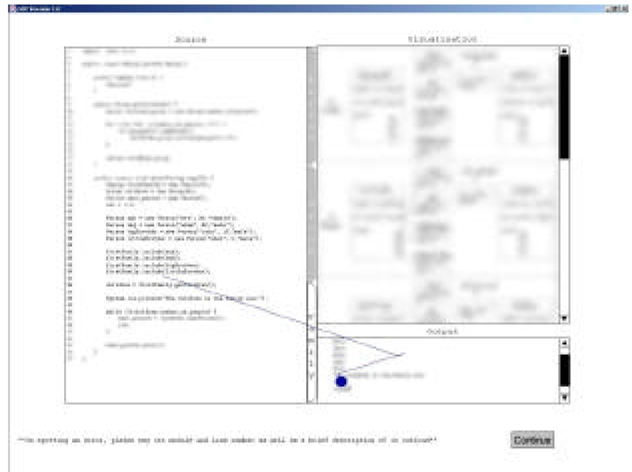

Figure 5. Debugging under RFV-on

REFERENCES

1. Bednarik, R., Tukiainen, M. Visual attention and representation switching in Java program debugging: A study using eye movement tracking. *16th Annual Psychology of Programming Interest Group Workshop (PPIG'04)*, pp.159-169, 2004.
2. Blackwell, A. F., Jansen, A.R., Marriott, K. Restricted Focus Viewer: A tool for tracking visual attention. In M. Anderson, P. Cheng & V. Haarslev (Eds.), *Theory and Applications of Diagrams*. Lecture Notes in Artificial Intelligence 1889, pp. 162-177, Springer Verlag, 2000.
3. Goldberg, J. H. and Kotval, X. P. Eye Movement-Based Evaluation of the Computer Interface. In Kumar, S. K. (Eds.), *Advances in Occupational Ergonomics and Safety*, Amsterdam: IOS Press, pp. 529-532, 1998.
4. Romero, P., Cox, R., du Boulay, B., Lutz, R. Visual attention and representation switching during Java program debugging: A study using the Restricted Focus Viewer. *Diagrammatic Representation and Inference: Second International Conference, Diagrams 2002* Callaway Gardens, USA. Lecture Notes in Artificial Intelligence, 2317, pp. 221-235, Springer Verlag, 2002.
5. Romero, P., Lutz, R., Cox, R., du Boulay, B. Co-ordination of multiple external representations during Java program debugging. *Empirical Studies of Programmers symposium of the IEEE Human Centric Computing Languages and Environments Symposia*, Arlington, VA, 2002.
6. Romero, P., du Boulay, B., Cox, R., Lutz, R. Java debugging strategies in multi-representational environments. *15th Annual Workshop of the Psychology of Programming Interest Group (PPIG'03)*, 2003.