A Method to Study Visual Attention Aspects of Collaboration: Eye-Tracking Pair Programmers Simultaneously

Sami Pietinen, Roman Bednarik, Tatiana Glotova, Vesa Tenhunen and Markku Tukiainen*
University of Joensuu

Abstract

The previous research of visual attention has mostly considered the situations in which a single person performs a task. The current eye-tracking devices and software support this research situation. Applications of eye-tracking in the research of collaborative tasks have been rare to date. We present a methodological framework of a research in which visual attention of pair programmers with a single display has been studied. We discuss the challenges of such research when conducted in real-world settings and the requirements on the eye-tracking setups. The hardware setups and software solutions to the problems of acquisition and synchronization of streams of eye-tracking data are presented. We outline the methodological questions of future visual attention research of collaborative tasks.

CR Categories: D.1.3 [Programming Techniques]: Concurrent Programming—Distributed programming;

Keywords: pair programming, eye-tracking, empirical research methodology

1 Introduction

The research in which eye-tracking has been applied to study collaborative programming has been rare to date. The present paper describes a research in which a complex cognitive environment is set up for a two-month-long software development project where two programmers worked as a pair within a larger, geographically distributed project team. The pair development was done using a single computer display screen and its contents were recorded and synchronized together with the eye movements of both programmers.

Setting up the research environment to acquire the visual attention data from two persons when both are making use of a single stimulus requires more effort than in the typical case of just one person. We report on the lessons learnt from setting up such an environment with the hope that such information can help the future eye-tracking research that studies the visual attention of two or more participants in a collaborative task.

2 Method Background

2.1 Pair Programming and Eye-Tracking

Pair programming (PP) is a method where two persons work together with an algorithm, design or programming task using one computer [Williams and Kessler 2000]. According to the literature, PP is said to yield benefits like less errors in code, simpler designs, more efficient problem solving, increased co-operation and communication, better productivity, and faster integration of newcomers into teamwork [Williams et al. 2000; Succi and Marchesi 2001; Williams and Kessler 2003]. Still, there have been only few empirical PP studies, and their results have been controversial about the claimed benefits; in particular, the alleged better quality and productivity have been questioned [Hulkko and Abrahamsson 2005]. On the other hand, some of the claims have been scientifically confirmed, including the better problem solving and learning.

To better understand and further improve the PP process, there is a need to investigate how the programmers do their work in the real world. When the programming task happens under the pair-programming paradigm, the joint shared use of the same information sources (as well as the use of the individual) might become one of the critical issues for a successful and effective performance. We made use of an eye-tracking system to record the programmers' visual attention.

To date there have not been eye-tracking studies reported that would investigate the aspects of tasks where two or more participants jointly collaborate and where visual attention would play an important role in the collaboration. We believe that one of the reasons behind this lack is a technical one. The restrictions of the eye-tracking technique have dictated the research setups and therefore the studies have mostly focused on single subject's visual attention. This seems to be the case also in the research where eye-tracking has been employed to study some aspects of programming (e.g. Stein and Brennan [2004]).

To study the collaborative aspects of PP at the moment, two separate eye-tracking cameras are needed to capture the eye movements of both programmers. It has been argued that a head-mounted camera – rather than a remote system – is easier to set up for the studies of programming [Nevalainen and Sajaniemi 2004]; however, it can be used only with a single person. To track the visual attention of the other programmer, there currently is a choice of a remote or a head mounted camera. As our goal was to combine both programmer's eye-tracking data with the single display screen, we found that the field of view of the head mounted camera was too large, and coordination of its output with the screen was too coarse. A remote eye-tracking system did not have this problem, although it allows the subject only minimal head movement. In addition, synchronization of the two streams and analysis of the resulting data presents a significant challenge. A more detailed description of the research environment is presented in section 3.1.

3 The Method

3.1 System Setup

After considering the possible combinations of the available devices, we ended up using the configuration as presented in Figure 1. The setup of the system allows recording of the following protocols: eye movement data of two users (one ASL 504 [ASL 2003b] protocol, one Tobii [Tobii 2003]), screen capture with cursor of the test-programs (video file), and facial videos of both users (two video files).

*e-mail: firstname.lastname@cs.joensuu.fi

ETRA 2008, Savannah, Georgia, March 26–28, 2008.
© 2008 ACM 978-1-59593-982-1/08/0003 $5.00

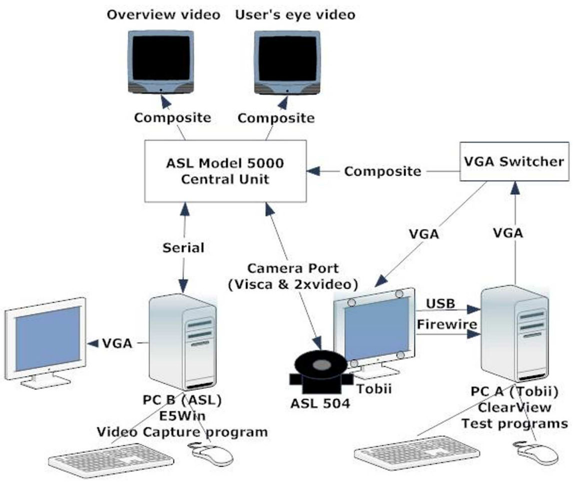

Figure 1: *System setup.*

System setup includes two PCs, one Tobii 1750 and one regular monitor, ASL 504, ASL Model 5000 Central Unit, VGA switcher, two monitors with composite input, cables, and two keyboards and mice (see Figure 1).

The PC A is responsible for running ClearView and recording eye-tracking from the Tobii monitor. Also, PC A runs the testing programs, the usage of which is a part of the original research problem to be analyzed. PC A is connected with the Tobii via USB and Firewire cables to transfer eye movement data from the Tobii, and through VGA Switcher via VGA cables to transfer VGA data from PC A to the Tobii system.

The PC B's only purpose is to record eye movement data from the ASL 504 and to control ASL 504 parameters, therefore it runs E5Win program. PC B is connected to the ASL Model 5000 Central Unit via serial cable in order to transfer data from the ASL 504. A regular monitor is used with PC B to start and stop the recordings and control the ASL 504.

The ASL Model 5000 Central Unit is responsible for processing the following data protocols:

- screen capture from PC A through VGA Switcher and via composite cable
- eye movement data and user's eye video from the ASL 504 via Visca and video cables

The Central Unit transfers this data to the following receivers:

- eye movement data to PC B via serial cable
- eye movement data and screen capture from PC A to Overview monitor
- user's eye video to User's eye video monitor
- controlling data from PC B to the ASL 504

In addition to these data protocols, we recorded facial video of both programmers. In the Tobii system a web-cam is attached to PC A via UBS cable, and ClearView saves the data. For recording of the facial expressions of the ASL 504 user, we used ASL 504 built-in camera. The data is transferred through the Central Unit and Overview video monitor's video output port to PC B. For saving the recording some other program has to be used rather than E5Win. The facial video protocol transfers and corresponding cables are not included in Figure 1.

One of the benefits of the described system setup is its robustness. If one of the computers would accidentally collapse, then only the protocol recorded with that computer is lost. The efficiency of this setup is higher compared to the case of using only one PC for running both recording programs and test-programs. Both of the recording programs demand great computing resources, and this would make a single PC too slow.

There are, however, some problems associated with this proposed system setup. These include the difficulty to achieve sustainable recordings of the ASL 504 user's eye movements. The eye should always be ”visible” to the ASL 504 camera, which restricts the head movements and makes it inconvenient for the user. Another problem concerns the synchronization of all recorded protocols, as every recording is started from different program and even from different computers. Extra additional tasks must be performed in order to achieve a correct synchronization, e.g. simultaneous calibration after the recording has started.

3.2 Recording and Calibration

3.2.1 Preparation of a Study

The analysis of very dynamic video content with large material base is very time consuming and even inaccurate related to automatic analysis. The content dynamics is not an issue with manual analysis, but it makes the automatization of analysis harder. Programming with Eclipse and similar environments has sources of dynamic content such as scrolling in code editor either vertically and horizontally, switching between different views for example when synchronizing with code repository, switching between tools and programs (e.g. from Eclipse and to an Internet browser). Minimizing the scrolling can be done by advising the participants to maximize the code editor window and by opening all the needed files at the start of the sofware development session.

Before we started the empirical part of this study, all the subjects were trained to use an agile software process called Mobile-D. This software development model is developed by our research partner. Mobile-D is based on Extreme Programming, Crystal methodologies and Rational Unified Process [Abrahamsson et al. 2004]. As an iterative development model, every iteration in Mobile-D consists of three types of days: planning day, working day, and release day. The day length is six hours and there is only four days in a week that are used for development, others are free of work.

An important aspect of conducting research in real-world settings is ergonomics. Because of the recording devices in this case, two programmers cannot be equally located in front of the monitor screen. The reviewer is located a little bit further to the left or right side of the monitor than the current programmer.

There are several ways to switch the roles when doing pair programming. The easiest way is to switch the mouse and keyboard and not to change the seats. It turned out to be difficult for participants to write with keyboard and simultaneously keep oneself in the boundary of the ASL 504 eye tracker. The problem of changing the roles resulted in a compromise: for longer sessions they just changed seats. This also meant that the recording devices had to be stopped and and they had to be calibrated for the other programmer. This was also a natural way to limit the file sizes of individual recordings.

3.2.2 Recording and Calibration Process

There are many way to conduct the calibration when two participants need to be calibrated. After testing several possibilities, we arrived at the following calibration procedure:

1. Calibrate the ASL if accuracy is not acceptable
2. Calibrate the Tobii if accuracy is not acceptable
3. Start recording in ClearView (Tobii)
4. Start recording in E5Win (ASL 504)
5. Ask both programmers to look at a same point on the screen
6. Ask them to look away from that point simultaneously
7. The actual session starts
8. After an hour, stop the recording and continue from step 1

4 Data and Analysis

4.1 Data Validity and Accuracy

The most notable issue to consider when doing eye-tracking to study programming is that with current devices, the accuracy of knowing where the programmer is looking at is physically limited to about a line of code. The precision of both the Tobii 1750 and the ASL 504 is 0.5 degrees from distance of 50 centimeters, which means an accuracy of 0.44 centimeters. Large head movements and long recording periods will decrease the accuracy. These limitation speaks to the behalf of AOI based analysis with bigger areas of interest. Additional techniques to improve the accuracy can be used, such as instructing the subject to speak out loud.

The accuracy and the amount of valid data of the eye trackers that we used have already been studied by Nevalainen and Sajaniemi [2004]. For the Tobii 1750, the mean distance was 1.134 centimeters from the actual requested point, which corresponds an angle 1.3 degrees from the distance of 50 centimeters. For the ASL 504 the mean distance from required point was 1.391, which corresponds the visual angle of 1.6 degrees using the same distance of 50 centimeters. In our study, the preliminary accuracy measurements seem to comply with these results.

The results for data validity in Nevalainen's and Sajaniemi's study were 8.1 percent of invalid data with the Tobii 1750 and 8.7 percent with the ASL 504. In our case the percentages presented for the ASL 504 cannot be reached, mainly due to two reasons. First, the location of the reviewer in respect to the monitor is not optimal for the eye tracker. Reviewer has to be located left to the monitor and the viewing angle comes very high compared to programmer that is almost in front of the monitor. Secondly, humans are known to express their emotions through body movement. In our case, the intensity of the collaboration between pair programmers is very high, which leads to high amount of body movements. These movements are hard to control in intensive situations, although this is highly dependent on the personal characteristics of the participants.

4.2 Eye Gaze Data Analysis

Analysis of eye gaze data can be done either manually or automatically. At this point, only manual analysis can be conducted with the video that contains both pairs' eye movements and the scene which is the video recorded from the monitor screen. On the other hand, if the gaze from different devices is analyzed with their own designated analysis programs, the whole process can be partially automated.

4.2.1 Manual Data Analyses

The analysis method to be used defines the way how to post-process the recorded eye gaze data. First, both of the pair programmers' eye movements have to be synchronized in the same video. ClearView version 2.6.3 records multiple protocols simultaneously, saving the screen capture video, eye gaze and monitor mounted web-cam video with sound into individual files. Combining these recorded materials can be done using ClearView with different video and audio encoders.

To obtain a video recording containing eye-movement traces of both programmers, we developed a tool called ETDPlayer [Rautiainen and Bednarik 2004]. The process of combining the two sources of eye-tracking data begins with the Tobii ClearView export. The recording contains the data of the first programmer (the driver). Next, we use the custom tool to combine and synchronize the ASL 504 eye-tracking data with the video exported from the Tobii ClearView software. For synchronization we use a point at which both programmers are looking at the same time. The ETDPlayer contains two timelines. One is used for the ASL-originated data and one for the video file (post-processed data from Tobii). The player allows adjusting the video file timeline along the ASL-based timeline.

Once the synchronization is done, the Tobii video file and the ASL data can be converted into a single video file, or visually analyzed directly in the ETDPlayer. The eye gaze data from ASL is not preprocessed like the Tobii data, and therefore more micro-movements can be seen (see Figure 2). The fixation points for the Tobii data are clear, but for the ASL data they are more difficult to figure out. We took two seconds time length for ASL eye gaze tracks and consider the concentration of ASL eye gaze tracks as fixations.

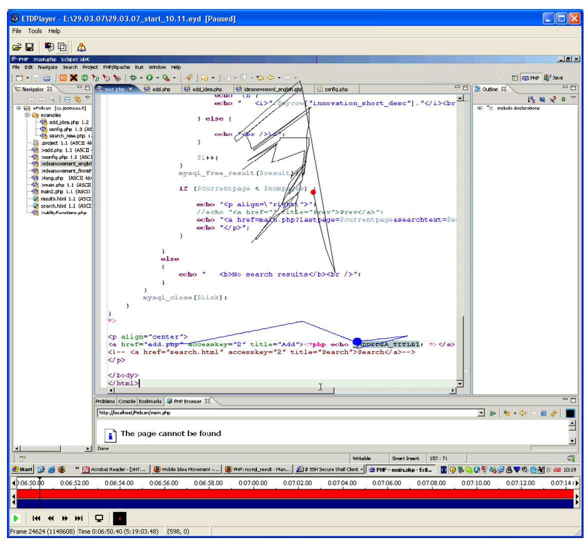

Figure 2: *The ASL and Tobii eye gaze data in one video.*

Manual visual data analysis, however, is time consuming process. In addition, the quality of ASL recordings compared to data produced by the Tobii is low. Therefore, we decided to employ also an AOI based data analysis, as described in the next section.

4.2.2 AOI Based Data Analysis

At the moment, there are no tools available to perform an analysis based on AOI (Area Of Interest) for combined data. Thus, in our study this type of analysis has to be conducted separately for both data sources. The recording containing both the eye-movement traces then serves as an exploratory material to identify interesting parts of the session. Those can then be analyzed in greater detail using the AOI based analysis. Because of the limitations caused by the instability of the ASL 504 recordings, the segments for the AOI based analysis are usually quite short and might have a negative impact on how representative the sample data are.

The AOI based data analysis can be automated with proper tools. Especially with large data sets, automatization of analysis is usually the only feasible and cost-effective method. Therefore, we plan to develop custom tools to conduct an AOI based analysis of visual attention data originating from multiple participants. For this purpose, a centralized protocol is needed to generate events that would facilitate the synchronization of the multiple eye-tracking data streams.

5 Discussion

The currently available eye-tracking technologies do not provide solutions for studies of collaborative aspects of visual attention. We presented one of many possible designs of such a system. While our solution is not necessarily novel, we believe that this report is useful for researchers wishing to study similar problems. The presented eye-tracking system setup is specific to the research of the collaborative programming and can be thought of as a starting point for future improvements.

While there is currently no ideal setup, most of the technical problems are minor and can be solved either by additional hardware tools or improved software. For example, to better integrate the two streams of eye-tracking data, an external timing server can be added to concurrently control and synchronize the recordings. A head-mounted magnetic tracker can be used to accommodate the head-movements of the programmer working as a reviewer. Finally, the problem of mutually incompatible file formats of eye-tracking devices can be solved by having an industry standard format.

The main challenges of the collaborative visual attention analysis are, however, related to the analysis of the data. The conventional eye-tracking measures such as the number of fixations (categorized into separate areas of interest), fixation times, and others can be used to describe the individual aspects of visual attention. The primary interest, however, is to try to find the differences, similarities or dependencies between the participants conducting a collaborative task and to compare them to the single-programmer situation.

One of the most important things in future research is to obtain more valid eye-tracking data. To this end, we will add a magnetig head tracker transmitter for the ASL 501, thus replacing the ASL 504. This enables us to fix the eye-tracker's target plane to the Tobii's display screen and discriminate most of the head movements. This way we will get data that can be directly combined with the Tobii's scene video. But even more importantly, this will improve the setup's usability by allowing programmers to move more freely and enabling a more realistic programming environment.

6 Conclusions

This paper describes a number of problems and their solutions when eye-tracking is used to study pair programming. The usage of two cameras to capture the eye movements of two programmers simultaneously in real software development tasks poses some new challenges compared to the traditional eye-tracking sessions with one individual in laboratory environment. The eye-tracking system must be unobtrusive, easy to use and its output needs to be easy to analyze.

We found that the available eye-tracking systems (head mounted, table mounted and display mounted devices) do not fulfill all of these requirements individually. It is possible, however, to create a satisfactory system with a combination of display mounted and table mounted cameras to capture sufficiently valid and good-quality data.

The current eye-tracking data combined as a single video creates a basis for qualitative analysis. Using the experiences gained, with an improved research setup we will be able to collect data which is also suitable for automatic quantitative analysis. Fo example, we can find out automatically how much the fixation points of the programmers are overlapping in order to separate the common visual attention interests. Again, AOI based analysis needs to be used, as the accuracy of eye-trackers is not yet high enough.

Still, most of the problems encountered when setting up the research environment were technical in nature, and therefore we believe they may be overcome with future advancements in eye-tracking technology.

References

ABRAHAMSSON, P., HANHINEVA, A., HULKKO, H., IHME, T., JÄÄLINOJA, J., KORKALA, M., KOSKELA, J., KYLLÖNEN, P., AND SALO, O. 2004. Mobile-d: an agile approach for mobile application development. In *OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, ACM, New York, NY, USA, 174–175.

ASL. 2003b. *Eye Tracking System Instruction Manual - Model 504 Pan/Tilt Optics*. Applied Science Laboratories.

HULKKO, H., AND ABRAHAMSSON, P. 2005. A multiple case study on the impact of pair programming on product quality. In *Proceedings of the 27th international conference on Software engineering*, 495–504.

NEVALAINEN, S., AND SAJANIEMI, J. 2004. Comparison of three eye tracking devices in psychology of programming research. In *Proceedings of 16th Annual Workshop of the Psychology of Programming Interest Group (PPIG'04)*, 151–158.

RAUTIAINEN, L. M., AND BEDNARIK, R. 2004. Etdplayer: a software to synchronize asl data files. Software. University of Joensuu.

STEIN, R., AND BRENNAN, S. E. 2004. Another person's eye gaze as a cue in solving programming problems. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, ACM, New York, NY, USA, 9–15.

SUCCI, G., AND MARCHESI, M. 2001. *Extreme Programming Examined*. Pearson Education.

TOBII. 2003. *Tobii 1750 Eye-tracker. Product description. Release B*. Tobii Technology AB.

WILLIAMS, L., AND KESSLER, R. 2000. All i really need to know about pair programming i learned in kindergarten. *Communications of the ACM 43*, 108–114.

WILLIAMS, L., AND KESSLER, R. 2003. *Pair Programming Illuminated*. Pearson Education.

WILLIAMS, L., KESSLER, R. R., CUNNINGHAM, W., AND JEFFRIES, R. 2000. Strengthening the case for pair programming. *IEEE Software 17*, 19–25.