Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Gaze Collaboration Patterns of Successful and Unsuccessful Programming Pairs

Maureen VILLAMOR[a,b*] & Ma. Mercedes RODRIGO[a]
[a]*Ateneo de Manila University, Quezon City, Philippines*
[b]*University of Southeastern Philippines, Davao City, Philippines*
*maui@usep.edu.ph

Abstract: In this paper, we characterize the gaze collaboration patterns of successful and unsuccessful programming pairs as they traced and debugged fragments of code. A dual eye tracking experiment was performed on pairs of novice programmers and their fixation sequences were analyzed using Cross-Recurrence Quantification Analysis (CRQA), which is an analysis on cross-recurrence plots (CRP). Other eye tracking metrics were also used. Findings revealed that successful and unsuccessful pairs can be characterized distinctively based on their CRQA results, CRPs, and other eye tracking metrics. Successful pairs have more incidences of low CRQA reflected on their CRPs as single and isolated points, presence of more white bands and empty regions, and few rectangular segments known as laminar or "trapped" states. On the other hand, the unsuccessful pairs have more occurrences of high CRQA manifested on their CRPs as heavily clustered points and visually recurring patterns because of their more pronounced similar scan and fixation cluster patterns compared to successful pairs. Other eye tracking metrics also provided differentiation between the successful and unsuccessful pairs. These preliminary findings provide the groundwork to objectively quantify and characterize collaboration among programming pairs and can also be used in similar studies to strengthen pair programming.

Keywords: Collaboration, pair programming, eye tracking, cross-recurrence plots

1. Introduction

In recent years, dual eye tracking has been used to study joint attention in pair programming (Pietinen et al., 2008; Jermann, Nüssli & Dillenbourg, 2011; Olsen et al., 2015). Two eye trackers can be utilized to study the gaze of two individuals working together to solve a problem (Pietinen et al., 2008). We begin this paper with the definitions of its three foundational concepts: the use of eye tracking to quantify collaboration, the concept of joint attention and its effect on collaboration, and pair programming.

Eye tracking is a technique whereby an individual's eye movements are measured so that the researcher knows where a person is looking at any given time, and how their eyes are moving from one location to another (Poole and Ball, 2006). Eye tracking methodologies have been applied to collaborative tasks either as a tool to understand interpersonal communication or describe how collaboration unfolds based on gaze patterns. For example, studies have investigated how quickly a test participant fixates on a target after it is mentioned by the partner (Richardson & Dale, 2005). These and similar measures indicate how well the listener understood what the partner said.

In collaborative learning situations, an indicator of productive collaboration is joint attention, i.e., "attending to something together with someone and being aware that both are attending" (Schilbach, 2015). Joint attention occurs when participants synchronize their gazes. Prior research has associated that the more joint visual attention partners share, the more productive collaboration often is (Jermann, Nüssli & Dillenbourg, 2011; Schneider & Pea; 2013).

Finally, pair programming is a form of collaborative learning where two programmers execute programming activities together. It may be co-located, where programmers share a single screen or may occur remotely or in a spatially distributed mode in which programmers look at the same code but on different screens (Baheti, 2002). It is a popular collaboration paradigm used in teaching introductory programming courses, with research showing benefits to students' learning

and attitudes towards programming, better quality of code, greater student confidence, increased likelihood of success in programming courses, faster completion of tasks, and attainment of goals that would seem difficult or impossible on an individual basis (Hannay, Arisholm, & Sjøberg, 2009; Olsen et al., 2015).

Eye tracking studies that used joint attention to assess collaboration in pair programming often employ the use of gaze coupling (Richardson & Dale, 2005), which refers to moments when the participants are looking at the same target. These studies claimed that eye gaze coupling could be an indicator of quality interaction and better comprehension (Richardson & Dale, 2005) and could reflect tightness of collaboration (Pietinen et al., 2008; Jermann, Nüssli & Dillenbourg, 2011). There is also evidence that moments of joint attention, particularly gaze patterns during program comprehension, are related to deeper and complex processing and the overall gaze coupling level is strongly related to the quality of the collaboration (Nüssli, 2011).

The goal of this paper is to characterize gaze collaboration patterns of novice programming pairs in the act of tracing fragments of code and debugging. To this end, we make use of an analysis method known as Cross-Recurrence Quantification Analysis (CRQA; to be discussed in greater detail in section 2). This paper attempts to answer the following: (1) Is there a significant difference on the CRQA results between successful and unsuccessful programming pairs? and (2) What characterizes the gaze collaboration patterns of successful and unsuccessful programming pairs using cross-recurrence plots and other eye tracking metrics? In this paper, the success of the collaboration is based on the debugging scores of the pairs. Our previous studies on the use of CRQA characterized gaze collaboration patterns according to the participants' prior knowledge (Villamor et al., 2017, Villamor & Rodrigo, 2017a), both prior knowledge and degree of acquaintanceship (Villamor & Rodrigo, 2017b), and determining leader-follower profiles (Villamor & Rodrigo, 2017c).

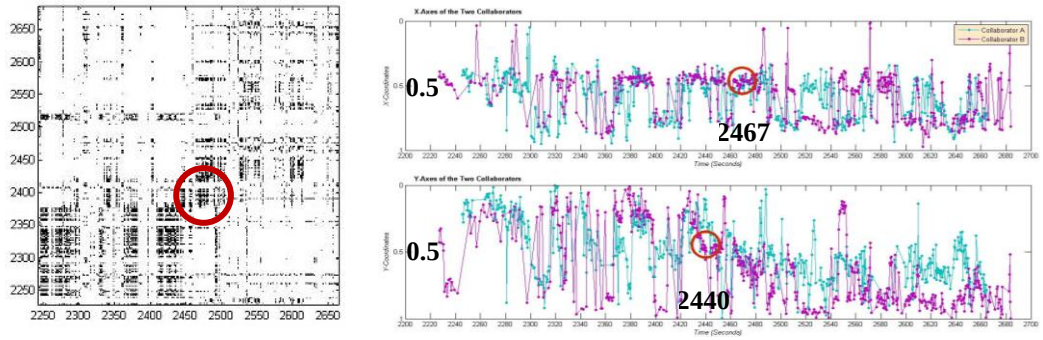
The primary goal of this research is to be able to build an explanatory model on the dynamics of pair programming as well as a predictive model capable of predicting the performance of the pairs using collaborators' profiles and behavioral indicators that can be automatically assessed and quantified. We intend to look at the pairs' collaboration as a whole as well as the individual differences of the collaborators within pairs and how these differences impact the success of the pairs. This study endeavors to contribute to this goal by investigating the coupling between the collaborators' gazes measured using CRQA to see whether the degree of coupling visualized by means of cross-recurrence plots (CRPs) can be used to distinguish successful and unsuccessful programming pairs. In addition to CRQA results and CRPs, other eye tracking metrics are also explored to provide support for the distinction between these programming pair categories.

2. Gaze Cross-Recurrence Plot

A cross-recurrence plot (CRP) is an N x N matrix, which is a representation of the time coupling between two time series. It is a form of a visualization which shows the simultaneous occurrence of similar states. Essentially, its purpose is to compare the states of two time series given the condition that the data should have the same unit and have the same phase space reconstruction. For example, in eye tracking, two fixation sequences from different collaborators, where each sequence contains the fixation x- and y-coordinates and the fixation timestamps can be used as the two time series. A cross-recurrence occurs when two fixations from different sequences land within a given threshold of each other using some distance metric (e.g., Euclidean distance). If fixations i and j are recurrent, they are represented as a black point or pixel on the plot. A CRP depicts fixations that are recurrent at their respective times. Figure 1.a shows an example of a CRP. The horizontal and vertical axes represent the time for the first and second collaborators, respectively. For example, both collaborators in Figure 1.a started at about the same time, which is about 2250 seconds past the starting time of the experiment.

On the CRP, different types of small-scale structures called textures may be identified (Marwan et al., 2007). Fading portions to the upper left and lower right corners mean that the data is non-stationary. Single and isolated recurrence points reflect random and strong fluctuations in the data. Horizontal and vertical lines and rectangular clusters denote that some states do not change or change very slowly for some time, which is an indication of laminar or "trapped" states. Bands of

white space indicate that transitions may have occurred and may reflect an underlying state change. This also means that the two collaborators uninterruptedly looked at two different spots on the screen. The empty regions indicate that the states within this period do not occur at any other times, which means that the states are unique. Diagonal lines parallel to the main diagonal of the plot means that a segment of one trajectory runs almost parallel to another segment. In eye tracking, this means that the collaborators looked at the same spot on the screen continuously.



(a) (b)

Figure 1. (a) An Example of a Cross-Recurrence Plot and (b) Scan Patterns using a Line Graph.

To understand better the relationship of CRPs and collaborative eye tracking, we demonstrate an example using one of the case scenarios in this study. Figure 2 shows snapshots of a program used as a stimulus in this experiment overlaid with colored circles. The colored circles are the fixation points of the two collaborators in this pair. The snapshot on the left with aqua-colored circles is for the first collaborator, and on the right with purple-colored circles is for the second collaborator. Above these snapshots are the times (in seconds) past the starting time of the eye tracking experiment when these fixations occurred. At these times, we can see how the fixation points are positioned at about the same location on the stimulus making these fixations recurrent based on a set threshold. In Figure 1.a, part of the pixelated regions enclosed in a red circle on the CRP informs us that the fixation points of the two collaborators under these specific times are recurrent.

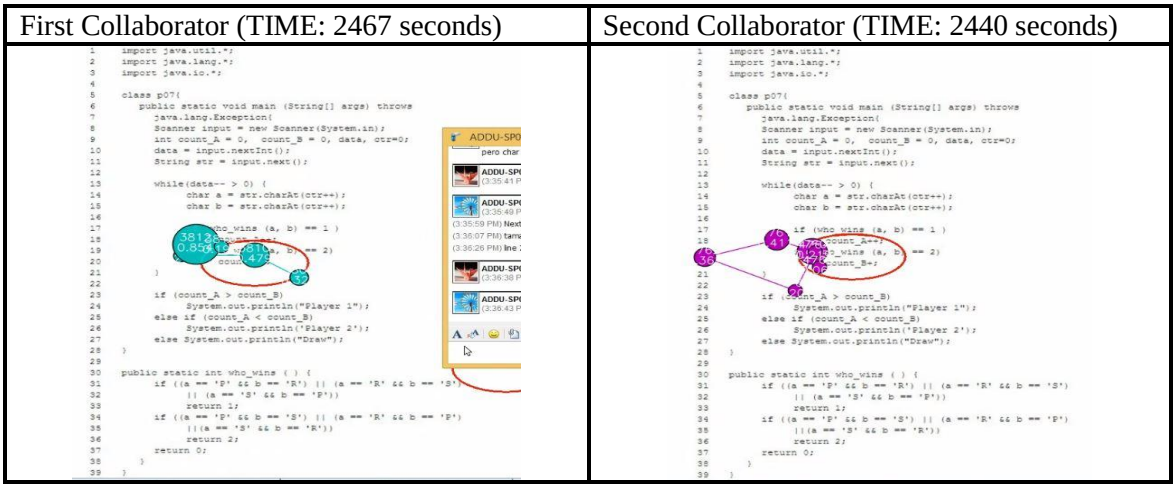


Figure 2. Snapshots of the location of the fixation points of the two collaborators.

Figure 1.b is the corresponding scan pattern using a line graph of the CRP in Figure 1.a. The two subplots illustrate the side-by-side comparison of the fixation *x*-coordinates (top subplot) and the fixation *y*-coordinates (bottom subplot) of the two collaborators. The aqua and purple line graphs refer to the first and second collaborators, respectively. The *x*-axes on the subplots represent the combined timeline of the two collaborators, and the *y*-axes represent the range of possible values of the fixation *x*- and *y*-coordinates, which is between 0 and 1 (reversed). The sections of the scan patterns enclosed in red circles in Figure 1.b show the positions in the timeline when these

fixations occurred. We can also see that the fixation *x*- and *y*-coordinates are positioned at about 0.5, which suggest that these fixation points from the two collaborators are indeed recurrent.

Analysis using CRPs is called Cross-Recurrence Quantification Analysis or CRQA (Zbilut, Guiliani & Webber, 1998). This determines how frequently two systems exhibit similar patterns of behavior over time by taking two different trajectories of the same information as input and performing a test for "closeness" between all the points of the two trajectories. This process is visualized using a CRP. Using CRQA, several metrics can be extracted from the diagonal and vertical dimensions of the CRP. These are recurrence rate, determinism, average and longest diagonal length, and entropy for the diagonal dimension; and laminarity and trapping time for the vertical dimension.

Cross-Recurrence Rate (*RR*) represents the "raw" amount of similarities between the trajectories of the two systems, which refers to the degree to which they tend to visit similar states. In eye tracking, this represents the percentage of cross-recurrent fixations that is indicative of the degree of gaze coupling or joint visual attention. Determinism (*DET*) is the proportion of recurrence points forming long diagonal structures of all recurrence points. Relative to eye tracking, this refers to the percentage of identical scanpath segments of a given minimal length in the two scanpaths.

The average diagonal length (*L*) reports the duration that both systems stay attuned. In eye tracking, this is the time where the scanpaths of the two collaborators run parallel for some time. The longest diagonal length (*LMAX*) denotes the longest uninterrupted period that both systems are in sync, which can be used as an indicator of stability of the coordination. In eye tracking, this represents the prolonged synchronization of the collaborators' scanpaths. Entropy (*ENTR*) measures the complexity of the attunement between systems. In eye tracking, this represents the complexity of the relation between scanpaths of the two eye movement data. *ENTR* is low if the diagonal lines tend to all have the same length, signifying that the attunement is regular; otherwise, *ENTR* is high if the attunement is complex.

Vertical structures in a CRP quantify the tendency of the trajectories to stay in the same region. The laminarity (*LAM*) of the interaction refers to the percentage of recurrence points forming vertical lines, whereas trapping time (*TT*) represents the average time two trajectories stay in the same region. In eye tracking, these metrics indicate the prolonged duration where the collaborators focus on certain regions of the screen, either to denote increased concentration or problems in comprehension. For a more detailed discussion of these metrics, refer to Marwan et al. (2007).

3. Methods

The study was conducted in 6 universities in the Philippines recruiting students who were in their 2nd to 4th year level in college and who had already taken their college-level fundamental programming course. Eighty-four (84) participants, 56 males and 28 females, were randomly paired regardless of gender and programming experience resulting in a total of 42 pairs. The task was to locate and mark the bugs in the 12 programs containing errors. The programs contained syntax, semantic, logic or a combination of these types of errors. Program complexity was categorized as easy, moderate, and hard depending on the type of errors the program contained. The distribution of the programs based on difficulty was as follows: easy (programs 1-3 and 10), moderate (programs 4-6 and 11), and hard (programs 7-9, and 12). Programs 1-3 contained a single bug and the rest had three bugs. The program comprehension test result was used to divide the participants into high and low proficiency levels. For a detailed description of the structure of the study and data cleaning and preparation, see Villamor and Rodrigo (2018).

A CRP was constructed for every program under each pair, and CRQA metrics were derived for each of the 12 programs. This was done using the CRP toolbox for MATLAB (Marwan et al., 2007). The CRQA parameters *delay* and *embed* were both set to one, and the *threshold* was set to a default of 10% of the maximal phase space diameter (Schinkel, Dimigen, & Marwan, 2008). Threshold adjustments were performed as needed due to varying fixation counts. This was to ensure that the threshold was neither too small nor too large. If the threshold is too small, the recurrence structure of the underlying trajectory may not provide us enough information. If the threshold is too large, almost every point is a neighbor of every other point, which could cause thicker and longer diagonal structures in the CRP as they actually are.

Two levels of granularity were used in the analysis: pair-level (average of all 12 programs) and case-level (all individual programs under each pair). A pair is successful if their average debugging score for the 12 programs is greater than or equal to the mean score; otherwise, the pair is unsuccessful. A case is successful when both participants within a pair are able to get at least half of the bugs in a program. Otherwise, if only one participant gets at least 50% of the bugs or both fail to spot the bugs, then the case is unsuccessful. A *t*-test for independent sample means at the 0.05 level of significance was performed to test for statistically significant differences on the CRQA results and other metrics between successful and unsuccessful pairs as well as successful and unsuccessful cases.

4. Results and Discussion

4.1 Differences in CRQA Results and CRPs

Of the 42 pairs, one pair was discarded due to huge fixation count discrepancies between collaborators, i.e., one had very high fixation count and the other had very low fixation count in all 12 programs. Other fixation sequences with very low fixation counts were not good candidates for CRQA and thus were not included. Hence, only 376 cases were used in this part of the analysis. Of the 41 pairs, there were 19 successful and 22 unsuccessful pairs. Of the 376 cases, there were 196 successful and 180 unsuccessful cases. Pair-level *t*-test results showed no significant differences on the CRQA results. On the case level, significant differences were found on the CRQA results between successful and unsuccessful cases at *RR* ($t = 6.981$, $p = 0.000$), *DET* ($t = 5.476$, $p = 0.000$), *L* ($t = 5.378$, $p = 0.000$), *LMAX* ($t = 3.435$, $p = 0.001$), *ENTR* ($t = 5.314$, $p = 0.000$), *LAM* ($t = 5.342$, $p = 0.000$), and *TT* ($t = 4.696$, $p = 0.000$).

Incidences of high and low instances of each CRQA metric in the successful and unsuccessful cases were examined to account for the differences between successful and unsuccessful cases. A CRQA value is high if it is equal to or greater than the mean plus one standard deviation; and low if it is equal to or lesser than the mean minus one standard deviation. Table 1 shows the descriptive values of the CRQA metrics, which covers the mean, standard deviation, the minimum and maximum values, and bases for low and high CRQA values. The large majority of the CRQA values in both categories were average. However, the successful and unsuccessful cases had more instances of low and high values, respectively, in all CRQA metrics. Figure 3 shows the percentage distribution of high and low CRQA values using a stacked column graph.

Table 1

Descriptive Values of the CRQA Metrics (N = 376)

CRQA Metric	Mean	Standard Deviation	Minimum	Maximum	Low <=	High >=
RR	0.40	0.13	0.04	0.76	0.27	0.54
DET	0.78	0.12	0.36	0.97	0.66	0.90
L	3.75	1.02	2.26	7.54	2.73	4.77
LMAX	27.44	17.13	5.00	111.00	10.31	44.58
ENTR	1.68	0.44	0.63	2.74	1.24	2.13
LAM	0.86	0.09	0.52	0.98	0.78	0.95
TT	5.23	1.82	2.34	5.23	3.41	7.04

Of the 196 successful cases, 125 cases or 63.78% were from the 19 successful pairs. Since this was more than the majority, we can characterize the successful pairs as having more incidences of low *RR*, low *DET*, low *L*, low *LMAX*, low *ENTR*, low *LAM*, and low *TT*. On the other hand, of the 180 unsuccessful cases, 112 of these or 62.22% were from the 22 unsuccessful pairs, which was also more than the majority. Hence, the unsuccessful pairs can be characterized as having more frequencies of high *RR*, high *DET*, high *L*, high *LMAX*, high *ENTR*, high *LAM*, and high *TT*.

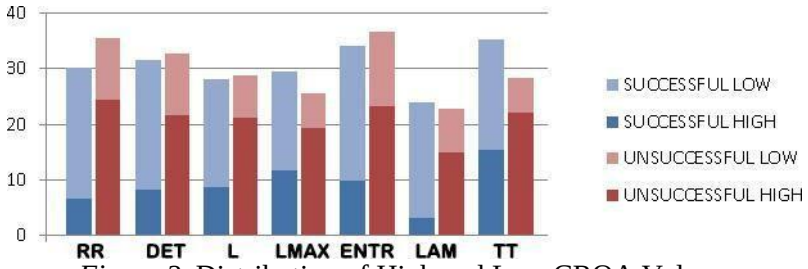


Figure 3. Distribution of High and Low CRQA Values.

The analysis was narrowed down on the 125 successful cases under successful pairs and the 112 unsuccessful cases under unsuccessful pairs. *T*-test results also revealed significant differences in *RR* ($t = 5.273$, $p = 0.000$), *DET* ($t = 4.455$, $p = 0.000$), *L* ($t = 3.727$, $p = 0.000$), *ENTR* ($t = 3.748$, $p = 0.000$), *LAM* ($t = 4.326$, $p = 0.000$), and *TT* ($t = 2.790$, $p = 0.006$) between these categories. The difference in *LMAX* was not significant. The majority of the significant CRQA values were still average, but based on the recomputed mean and standard deviation, the successful pairs indeed had more occurrences of low *RR*, low *DET*, low *L*, low *ENTR*, and low *LAM*. However, the number of high *LMAX* values was greater than the low *LMAX* values; and the number of high and low values for *TT* were comparable. The unsuccessful pairs also proved to have more values of high *RR*, high *DET*, high *L*, high *LMAX*, high *ENTR*, high *LAM*, and high *TT*.

To determine further what factors could have contributed to the CRQA differences, we extracted all successful pairs (i.e., successful cases under successful pairs) that fit the following criteria: low *RR*, low *DET*, low *L*, low *LMAX*, low *ENTR*, low *LAM*, and low *TT*. We also extracted unsuccessful pairs (i.e., unsuccessful cases under unsuccessful pairs) with high *RR*, high *DET*, high *L*, high *LMAX*, high *ENTR*, high *LAM*, and high *TT*. Fifteen (15) successful pairs and eleven (11) unsuccessful pairs fit the criteria. The CRPs were examined to draw out observable differences that could potentially explain the characterizations between successful and unsuccessful pairs. We found that all 15 successful pairs had fixation counts which were either low or below the average. All but one of the 11 unsuccessful pairs had fixation counts which were either high or above the average. The low fixation counts could have reduced the possibility of obtaining more recurrent fixations, and hence, resulted to more "low *RR*" in successful pairs; whereas the high fixation counts could have increased the likelihood of having more recurrent fixations, which led to more "high *RR*" in unsuccessful pairs.

Disparities in the CRPs of the successful and unsuccessful pairs were evident. Figure 4 shows an example of a CRP, scan pattern, and a pair fixation map of one of the successful pairs. The low fixation count was depicted on the CRP as mostly single and isolated points and more bands of white spaces and empty regions denoting abrupt transitions or state changes. The fixations also appeared to be more dispersed as seen on the fixation map and reflected on the scan pattern. On the other hand, an example of a CRP, scan pattern, and pair fixation map of one of the unsuccessful pairs is shown in Figure 5. The high fixation counts of the unsuccessful pairs resulted to having rectangular or larger clusters of points. Visually noticeable recurring patterns were also shown on most of the CRPs of the unsuccessful pairs, which were not apparent on the CRPs of the successful pairs. It is possible that the heavily pixelated regions and the visual recurring patterns found on the CRPs of the unsuccessful pairs is a result of a high degree of gaze coupling or cross-recurrent fixations.

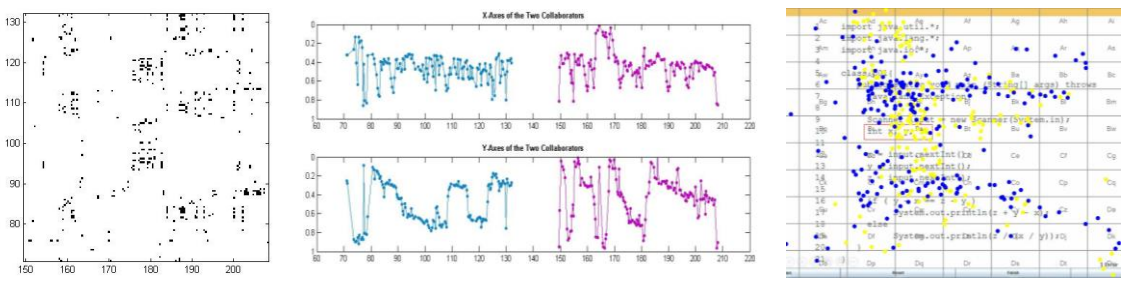


Figure 4. CRP, scan pattern, and pair fixation map of one of the successful pairs.

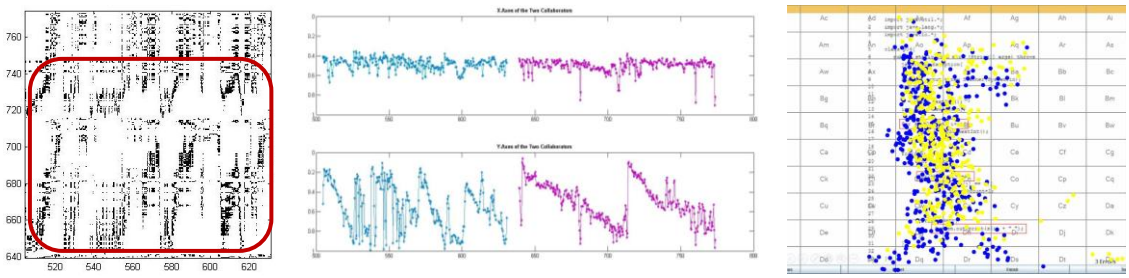


Figure 5. CRP, scan pattern, and pair fixation map of one of the unsuccessful pairs.

It was also observed that the scan pattern similarities of the unsuccessful pairs were more pronounced when compared to the successful pairs. Their fixations also tended to cluster on the same area on the screen or same locations in the program as seen on the pair fixation map in Figure 5. These scan pattern similarities and fixation cluster patterns consequently resulted to a higher degree of gaze coupling causing more incidences of high *RR*s in the unsuccessful pairs. In short, the high fixation counts, high degree of scan pattern similarities, and fixation clusters contributed to more high *RR*s, which is a characteristic of unsuccessful pairs. *RR* happens to be highly correlated to *DET* ($r = 0.904$, $p = 0.000$), *L* ($r = 0.904$, $p = 0.000$), *LMAX* ($r = 0.785$, $p = 0.000$), *ENTR* ($r = 0.923$, $p = 0.000$), *LAM* ($r = 0.828$, $p = 0.000$), and *TT* ($r = 0.852$, $p = 0.000$), and hence, when *RR* increases, these other CRQA metrics follow suit.

Is it possible that the number of bugs, program complexity, and the proficiency of students confounded the results? Upon inspection, we found that 11 and 4 out of the 15 successful cases representing the successful pairs came from programs that were categorized as easy (with single bug) and moderate (three bugs), respectively. On the other hand, 7 and 4 out of the 11 unsuccessful cases representing the unsuccessful pairs emanated from programs tagged as hard (three bugs) and moderate (three bugs), respectively. Given a single bug to locate, the program difficulty level, and general overall proficiency of the successful pairs, it is expected that they would find the bug easily. On the other hand, with three errors to uncover from both moderate and hard programs, the unsuccessful pairs composed mostly of low proficiency students, would likely have difficulty spotting the bugs. Hence, to control for these confounds, successful cases which were representative of the hard programs were selected as well as unsuccessful cases that were representative of the easy programs containing only a single bug.

Prior to examining their CRPs and scan patterns, two separate *t*-tests were performed to determine if there were significant differences on the CRQA values between successful and unsuccessful cases on easy and hard programs. There were 99 out of the 376 cases that belong to easy programs, where 53 of these were successful and 46 were unsuccessful. Significant differences were found in terms of *RR* ($t = 4.401$, $p = 0.000$), *DET* ($t = 3.577$, $p = 0.001$), *L* ($t = 3.330$, $p = 0.001$), *LMAX* ($t = 2.303$ $p = 0.023$), *ENTR* ($t = 3.458$, $p = 0.001$), *LAM* ($t = 3.239$, $p = 0.002$), and *TT* ($t = 2.704$, $p = 0.008$) between successful and unsuccessful cases. On the other hand, there were 133 cases that belonged to hard programs, where 61 of these were successful and 72 were unsuccessful. Significant differences were also found in *RR* ($t = 4.020$, $p = 0.000$), *DET* ($t = 3.022$, $p = 0.003$), *L* ($t = 3.124$, $p = 0.002$), *ENTR* ($t = 2.975$, $p = 0.003$), *LAM* ($t = 3.436$, $p = 0.001$), and *TT* ($t = 2.781$, $p = 0.006$) between successful and unsuccessful cases. The difference in *LMAX* was not significant.

Eight (8) CRPs each of the successful cases representative of hard programs and unsuccessful cases representative of easy programs with a single bug were sampled. While the single and isolated points and the presence of more bands of white spaces and empty regions were still evident on the CRPs of the successful pairs, small clusters of points were already seen forming mostly in vertical and horizontal patterns, which are indications of laminar phases. More incidences of these so-called laminar states increase the value of *LAM* and could also result to a high *TT*. The presence of these laminar states on the CRPs of the successful pairs suggests that they also need more time to locate all three errors by concentrating on certain locations where bugs are more likely to occur. Nonetheless, these clusters of points were larger and were more prominent in the unsuccessful pairs, which could also explain why they had more high *LAM*s and high *TT*s. The high *LAM*s and high *TT*s could be the reason also for the visually recurring patterns found on the CRPs of

the unsuccessful pairs. Referring to the scan patterns of the successful pairs, we also observed that the more similar the scan patterns, the more pronounced the pixelated regions were.

Conversely, there were also isolated incidences of recurrence points found on the CRPs of the unsuccessful pairs, but the points were still heavily clustered in most of the CRPs. This could mean that given only a single error to locate, the unsuccessful pairs would still have difficulty in finding this single error in the program because they need more time to spot the error. The scan patterns of the unsuccessful pairs also seemed to be more similar independent of the time when these fixations happened. This further suggests that unsuccessful pairs could be following a certain program scanning pattern in finding bugs.

4.2 Differences using other eye tracking metrics

Included in this analysis were all the 376 cases, where the 196 and 180 of these were successful and unsuccessful cases, respectively. The metrics we used for this characterization were the following: loci similarity, sequence similarity, duration per program, total fixation count per stimulus and per area of interest (AOI), total fixation time per AOI, time to first fixation per AOI, and average fixation duration per AOI. These metrics were calculated, and the erroneous lines of code were converted to AOIs using the OGAMA software (Voßkühler et al., 2008).

Levenshtein distance, also known as edit distance algorithm (Levenshtein, 2002), was used in the computation for the loci similarity and sequence similarity between the collaborators' scanpaths. Loci similarity refers to the percentage of locations both scanpaths have passed by, independently of time and sequence. A loci similarity value of 100% denotes that the collaborators' fixations where at the same locations on the stimulus, whereas a sequence similarity value of 100% means that the collaborators have identical scanpaths (Voßkühler et al., 2008). To do this, each stimulus was divided into a 10 x 10 grid, where each cell in the grid was assigned a unique letter. A scanpath was built into a string using the letters of the cell that contained the current fixation location. Examples of this grid can be seen on the pair fixation maps in Figures 4 and 5. Levenshtein distance was then applied by counting the number of operations (deletions, insertions, substitutions) needed to transform one string into the other. For the other metrics, the average values of the two collaborators within pairs were used.

The results of these metrics are shown in Table 2. In the discussion that follows, we will refer to successful cases as successful pairs and unsuccessful cases as unsuccessful pairs. The unsuccessful pairs have significantly higher loci and sequence similarity than successful pairs. These results confirmed the incidences of more "high *RR* and *DET"* values in the unsuccessful pairs. The duration per program between the successful and unsuccessful pairs was not significant.

Table 2

Descriptive Values of the other Metrics (time is measured in seconds)

Metric	Mean		Standard Deviation		*t*-value	*p*-value
	S	UN	S	UN		
Loci similarity	62.41	64.60	9.96	10.06	2.114	0.035
Sequence similarity	12.13	13.00	4.29	4.12	2.008	0.045
Duration per program	1062.32	974.25	737.53	679.22	Not significant	
Total fixation count (stimulus)	579	604	376	324	Not significant	
Fixation count (AOI)	66	86	58	62	3.165	0.002
Total fixation time (AOI)	19.35	26.91	16.14	26.55	3.367	0.001
Time to first fixation (AOI)	70.20	61.02	180.46	139.61	Not significant	
Fixation duration mean (AOI)	0.30	0.32	0.07	0.27	Not significant	

The total fixation count per stimulus was not significant but the fixation count per AOI was significant, with the successful pairs having lower fixation count per AOI than the unsuccessful pairs. Since most of the successful pairs were highly proficient, this result is in line with prior research findings that highly proficient participants have lower fixation counts than participants with low proficiency level. Successful pairs also had significantly shorter fixation times per AOI. This implies

that once the successful pairs spot already the error/s and both agree that it is indeed an error, they will no longer spend more time on that AOI and will transition to next AOI or program. Hence, the successful pairs are more confident with their answers. The unsuccessful pairs, on the other hand, may be plagued often with uncertainty or lack of confidence with their answers causing them to spend more time on deciding if that particular line or program location contains the erroneous lines of codes. Lastly, the time to first fixation and fixation duration mean per AOI were not significant.

5. Implications

Attrition rate in introductory programming is known to be high so teachers strategize using various methods to reduce this. One of these popular strategies is to engage their students in collaborative learning tasks such as pair programming. As per Schneider and Pea (2013), mutuality of exchanges and the degree of joint attention are determinants of a successful collaboration. The outcome of collaboration does not solely depend on the contributions of the individuals but also on how efficiently group members manage individual and joint attention during collaborative tasks. Hence, if the concept of both individual and joint attention can be explored further, this can be used to improve the quality of collaboration in programming pairs. Joint attention can be made intentional, and thus, can be increased by encouraging the pairs to connect in more conversational processes that will result to a more productive collaboration. This study also emphasizes the importance of collaboration and provides a precursor on ways to objectively quantify and characterize collaboration among programming pairs. Since this study provides the groundwork to distinguish between successful and unsuccessful pairs, this gives us the impression that we can learn how successful pairs collaborate and identify what factors make them successful so that others who are struggling in programming can do the same.

6. Summary, Conclusion, Limitation, and Future Work

This paper characterized the gaze collaboration patterns of successful and unsuccessful programming pairs using cross-recurrence plots and its associated metrics as well as other eye tracking metrics. Findings revealed that the difference on the CRQA results between successful and unsuccessful pairs are significant. The successful pairs are characterized as having lower fixation counts on predefined AOIs and more incidences of low CRQA values. The CRPs of the successful pairs can be described mostly as having more single and isolated points, presence of more bands of white spaces and empty regions, and few rectangular segments of recurrence points. These characterizations are indications that successful pairs finish faster, have more preference for independent work at certain times, may have shared similar but shorter scanpaths frequently, have more frequent scan path transitions, and transition faster so they find bugs quickly. On the other hand, the unsuccessful pairs are characterized as having higher fixation counts on predefined AOIs and more occurrences of high CRQA values. Their CRPs have evidence of heavily clustered recurrence points or larger laminar states and visually recurring patterns. These suggest that unsuccessful pairs need lengthy consideration of the program, may have shared similar scanpaths that are longer, follow a certain pattern in locating bugs, may look at the same area repeatedly because they do not know where else to look, use trial-and-error in debugging, and may usually have problems in program comprehension.

Although the concept of joint attention has been found to be an indicator of productive collaboration, this study however lacks the internal validity to make any causal claims about the relation between joint attention and programming performance of the pairs. On the question as to which one collaborated better, it is still premature to conclude that unsuccessful pairs collaborated better than successful pairs based on the degree of gaze coupling alone. If one pair has a high *RR* and another has a low *RR*, it does not necessarily follow that the former is more coupled, but it may mean that they could be working on a smaller area of the screen. The high degree of gaze coupling in the unsuccessful pairs could also just be a result of an unintentional gaze coordination, as opposed to a gaze coordination that is generated by conversational processes. Hence, to account for this, other pair dynamics will be investigated as well as the nature of the discourse. As this is just a preliminary

study, other factors (e.g., individuals within pairs, participant profiles, conversational processes, etc.) will also be investigated in the future and consider other methodologies to develop a better understanding of pair programming and how collaboration impacts the pairs.

References

Baheti, P. (2002). Assessing distributed pair programming. In *Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 50-51).

Hannay, J. E., Dybå, T., Arisholm, E., & Sjøberg, D. I. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, *51*(7), 1110-1122.

Jermann, P., Mullins, D., Nüssli, M. A., & Dillenbourg, P. (2011). Collaborative gaze footprints: Correlates of interaction quality. In *Connecting Computer-Supported Collaborative Learning to Policy and Practice: CSCL2011 Conference Proceedings.* (Vol. 1, No. EPFL-CONF-170043, pp. 184-191). International Society of the Learning Sciences.

Levenshtein, V. I. (2002). Bounds for deletion/insertion correcting codes. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on* (p. 370). IEEE.

Marwan, N., Romano, M. C., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics reports*, *438*(5-6), 237-329.

Nüssli, M. A. (2011). *Dual eye-tracking methods for the study of remote collaborative problem solving* (Doctoral dissertation, École Polytechnique Fédérale de Lausanne).

Olsen, J. K., Ringenberg, M., Aleven, V., & Rummel, N. (2015). Dual eye tracking as a tool to assess collaboration. In *ISLG 2015 fourth workshop on intelligent support for learning in groups* (pp. 25-30).

Pietinen, S., Bednarik, R., Glotova, T., Tenhunen, V., & Tukiainen, M. (2008, March). A method to study visual attention aspects of collaboration: eye-tracking pair programmers simultaneously. In *Proceedings of the 2008 symposium on Eye tracking research & applications* (pp. 39-42). ACM.

Poole, A., & Ball, L. J. (2006). Eye tracking in HCI and usability research. *Encyclopedia of human computer interaction*, *1*, 211-219.

Richardson, D. C., & Dale, R. (2005). Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive science*, *29*(6), 1045-1060.

Schilbach, L. (2015). Eye to eye, face to face and brain to brain: novel approaches to study the behavioral dynamics and neural mechanisms of social interactions. *Current Opinion in Behavioral Sciences*, *3*, 130-135.

Schinkel, S., Dimigen, O., & Marwan, N. (2008). Selection of recurrence threshold for signal detection. *The European Physical Journal-Special Topics*, *164*(1), 45-53.

Schneider, B., & Pea, R. (2013). Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-supported collaborative learning*, *8*(4), 375-397.

Villamor, M., Paredes, Y. V., Samaco, J. D., Cortez, J. F., Martinez, J., & Rodrigo, M. M. (2017). Assessing the Collaboration Quality in the Pair Program Tracing and Debugging Eye-Tracking Experiment. In *International Conference on Artificial Intelligence in Education* (pp. 574-577). Springer, Cham.

Villamor, M. M., & Rodrigo, M. M. T. (2017). Characterizing Collaboration in the Pair Program Tracing and Debugging Eye-Tracking Experiment: A Preliminary Analysis. In *Proceedings of the 10th International Conference on Educational Data Mining* (pp. 174-179).

Villamor, M. and Rodrigo, M. M. (2017) Impact of Both Prior Knowledge and Acquaintanceship on Collaboration and Performance: A Pair Program Tracing and Debugging Eye-Tracking Experiment. In *Proceedings of the 25th International Conference in Computers in Education* (pp. 186-191).

Villamor, M. and Rodrigo, M. M. (2017) Exploring Lag Times in a Pair Program Tracing and Debugging Eye-Tracking Experiment. In *Proceedings of the 25th International Conference in Computers in Education* (pp. 234-236).

Villamor, M. and Rodrigo, M. M. (2018). Predicting Successful Collaboration in a Pair Programming Eye Tracking Experiment. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM.

Voßkühler, A., Nordmeier, V., Kuchinke, L., & Jacobs, A. M. (2008). OGAMA (Open Gaze and Mouse Analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behavior research methods*, *40*(4), 1150-1162.

Zbilut, J. P., Giuliani, A., & Webber Jr, C. L. (1998). Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, *246*(1-2), 122-128.