Collaborative Gaze Footprints: Correlates of Interaction Quality

Patrick Jermann, CRAFT, Ecole Polytechnique Fédérale de Lausanne (EPFL), Patrick.Jermann@epfl.ch
Dejana Mullins, University of Freiburg, D-79085 Freiburg, dejana.mullins@gmail.com
Marc-Antoine Nüssli, Pierre Dillenbourg, CRAFT, Ecole Polytechnique Fédérale de Lausanne (EPFL),
Email: Marc-Antoine.Nuessli@epfl.ch, Pierre.Dillenbourg@epfl.ch

Abstract: Dual eye tracking offers new possibilities for the analysis and diagnosis of collaborative interaction. Cross-recurrence analyses and visualizations offer insight into how closely two collaborators' gaze follow each other. We contrast two cases to illustrate how gaze cross-recurrence can be used as a correlate of high and low quality interaction. The intriguing graphical patterns that result from the time coupled traces of the collaborators' fixations are footprints of the quality of the interaction. Good quality interaction features a higher recurrence rate than low quality interaction. The graphical structure of the recurrence plots indicates whether collaborators divide labor and whether they are sharing visual attention.

Introduction

The long-term goal we pursue is to use behavioral indicators that can be measured and evaluated automatically to represent the quality of interaction along psychologically meaningful dimensions. These dimensions could then be used as a basis to diagnose the quality of interaction (Soller, Martinez, Jermann & Mühlenbrock, 2005).

We contribute to this endeavor by pointing out gaze-based correlates of rating dimensions that are used in CSCL to characterize the quality of collaborative interaction (Meier, Spada, & Rummel, 2007). The assumption behind this approach is that the type of coupling between collaborator's gaze reflects the quality of their interaction. We focus in this contribution on cross-recurrence, a measure of whether two collaborators look at the same target more or less at the same time (typically within a 4 seconds time span). We see cross-recurrence visualizations (Figure 1) as footprints of the quality of interaction that allow to visually grasp the gist of the interaction dynamics. More generally, our approach aims at describing and modeling collaboration across several levels of control (Lord & Levy, 1994), from low-level individual signals up to social interaction via an analysis of synchronization and alignment between individual behaviors.

Dual Eye Tracking and Gaze Recurrence

Synchronous dual eye tracking is a novel methodology, which consists of recording the gaze of two collaborators simultaneously (see for instance Cherubini, Nüssli & Dillenbourg, 2008 and Nüssli, Jermann, & Dillenbourg, 2009). Existing research about gaze and communication studied situations that are often about "delayed" communication (subjects describe a scene for some listener who is not present) and about simple visual referents (the scene looked at by the subjects is simple and static). Richardson and Dale (2005) for example used eye movements to investigate how a speaker and a listener deployed their attention within a visual "common ground". Their research focused on cases where two partners are looking at a visual scene that is the topic of conversation. A cross-recurrence analysis showed that speaker and listener eye movements were coupled throughout the discourse. The more closely a listener's eye movements were coupled with the speaker's, the better the listener did on comprehension questions. In this study, speaker and listener were not interacting synchronously. The result was however replicated in a study with synchronous dialogue (Richardson, Dale, & Kirkham, 2007). Cross recurrence quantification analysis was proposed recently as a generalized method to unveil the interlocking of two interacting people (Dale, Warlaumont & Richardson, in press). The authors used the method to analyze gaze as well as the use of language alignment through grammatical sequences (Dale & Spivey, 2006). In previous studies conducted in our lab, we found that gaze proximity was, at the micro-level, negatively related to the misunderstandings of referential utterances (Cherubini et al., 2008) and, that cross-recurrence was at the macro-level, positively correlated to team performance (Cherubini, Nüssli & Dillenbourg, 2010). Some initial observations by Pietinen and colleagues (2008) in the domain of pair programming also suggest that gaze closeness might reflect tightness of collaboration. The driver and navigator roles that are typical for pair programming seem also to be reflected in the way programmers look at the code.

We extend the work by these authors on pair programming, a task that is more complex than controlled referential communication and more realistic than faked interaction. The notion of footprint that we use in this contribution is related to the notion of "signature" that is used by Dale et al. (submitted) to refer to the high level visual patterns (lines and structures) in cross-recurrence plots. In physics, where the cross-recurrence plots initially stem from, Eckman Kamphorst, and Ruelle (1987) also describe two levels of reading recurrence plots: "large scale typology" which refers to the overall aspect of the plots as containing structure or being homogeneous; and "small scale texture" which refers to smaller details and reflects intrinsic properties of the dynamical systems represented.

Collaboration Quality

CSCL encompasses diverse approaches for analyzing interaction that differ in the purpose of the analysis, the units of interaction that are analyzed, and the underlying theoretical assumptions. In a recent workshop series (e.g. Suthers, Law, Lund, Rose, & Teplovs, 2009), this diversity and the importance of achieving a common ground in the analysis of interaction was specifically addressed.

Traditionally, researchers applied a "coding and counting" approach in order to evaluate the effectiveness of collaboration processes (e.g. De Wever, Schellens, Valcke, & Van Keer, 2006; Strijbos, Marten, Prins, & Jochems, 2006), that is, they transcribed the interaction and counted the occurence of particular types of utterances. This approach is not only very time consuming, but the results of a coding analysis also do not necessarily inform about the success of the collaboration. For instance, a high amount of utterances concerning the technical coordination can also indicate that the collaboration partners have difficulties in coordinating the use of the technical environment. Meier et al. (2007) therefore suggested employing a rating analysis. They developed a rating scheme to analyze the quality of collaborative interaction. The rating scheme contained nine dimensions that evaluated aspects of communication, joint information processing, coordination, interpersonal relationship, and motivation. The evaluation of the rating scheme testified a high inter-rater reliability, consistency, and validity. Originally, the rating scheme was developed for the context of interdisciplinary collaboration. However, Rummel, Deiglmayr, Spada, Kahrimanis, and Avouris (2011) showed that it can also be adapted for other contexts, for instance for the analysis of collaborative interaction between computer scientists, or for the analysis of students' collaborative learning in mathematics (see also Diziol et al., 2008).

Research Question

The goal of this contribution is to point out gaze-based correlates of the dimensions of collaborative interaction identified by Meier et al. (2007). To this end, we present a case study that aims at identifying the characteristics of gaze traces that are related with interaction quality. In order to identify relevant gaze features, we contrasted two cases: One dyad that showed high collaboration quality and one dyad that showed low collaboration quality based on the interaction analysis with a rating scheme. By choosing diverse examples, we exploit the natural variability between the cases to identify relevant gaze patterns (Firestone, 1993). Our main question is what are the differences in the gaze patterns of the good and the bad dyad, and how do these differences relate to the interaction quality?

Method

We report data that was collected during the piloting phase of a larger study of program understanding that involved 42 pairs and 55 individuals, all students in computer science and communication science, and which is under analysis at the time of writing. We focus specifically in this contribution on three master students in computer science in their 4th and 5th year and one bachelor 3rd year student in communication science. They all report to program at least between 1 to 3 hours per week in Java. We refer to the participants as "Blue" and "Red". In Dyad 1, Blue indicated that he has "poor" knowledge of Java, whereas Red indicated that he has "good" knowledge. In Dyad 2, Blue indicated "good" knowledge and Red indicated "very good" knowledge. Despite this self-reported disparity of expertise, the dyad partners had comparable prior knowledge as assessed in a pre-test: In both dyads, one student reached a prior knowledge score of 5 out of 13 points (Blue in both dyads), and the other student reached a prior knowledge score of 7 points (Red in both dyads).

Dyads worked on a task that consisted of describing the rules of a game implemented as a Java program of three hundred lines. Understanding and explaining the rules of the game required subjects to translate programmatic Java code into a domain model that is expressed in everyday language. In the game, two players alternated in choosing numbers out of a list of nine numbers from 1 to 9. The goal of the game was to have three numbers that add up to 15. The instructions for the dyads were: "You have been hired in a game company and you have received this code. This is a two player turn-based arithmetic game. Please explain the rules to potential non-programmer players including: What is the initial game situation, what does a player have to do at each turn. Give an example of a valid and an invalid action. What are the general rules of validity of an action? What does a player have to do or get in order to win the game? Under which condition can the game end without a winner?"

The experiment consisted of two phases. First, students completed an individual pre-test to assess their prior knowledge in programming and more specifically their java skills (individual phase). Then they collaboratively solved ten programming tasks (collaborative phase). During the collaboration, the dyad partners were seated in separate rooms, but were able to communicate with each other through an audio channel. Each question was timed, and the dyads received a warning 20 to 45 seconds before time ran out. Although we recorded the entire interaction, the dyads were asked to indicate when they were ready to answer to the tasks by clicking a button labeled “Start recording”. The entire experiment lasted for approximately 90 minutes.

The problem-solving environment consisted of a collaborative programming editor based on the Eclipse IDE (http://www.eclipse.org). The workspace consisted of an instruction view (Instructions area) at the top of the screen that contained the question the subjects had to answer, and a central view (Code area) displaying the code of the program. The lower part of the editor displayed the remaining time and contained a button for recording the answer (Time/Management area). The workspace was shared via a server that relayed interface events to connected clients, i.e. if one partner used the scroll bar, the view of his partner changed as well to ensure that they were viewing the same section of the code. Furthermore, the dyad partners were able to highlight parts of the code in order to draw the partner’s attention to specific aspects or sections. Two Tobii 1750 eye-trackers were used to record gaze at 50 Hz. A dedicated server synchronized the eye-trackers’ recordings, and the data was logged using callback functions from low-level API of the eye-trackers. The subjects’ heads were held still with a chinrest placed at 65 centimeters of the screen. An adaptive algorithm was used to identify fixations and a post-calibration was done to best align fixations with the stimulus. While subjects used Skype to converse, audio was recorded using dedicated phantom powered boundary microphones amplified through a preamplifier.

Analysis of Collaboration Quality

To evaluate the collaboration quality of dyadic pair programming, we replayed the interaction of the dyads using an interactive tool, which displays the shared screen, the selections done by the subjects, and plays back their conversation and eye gaze. The analysis was two-folded: First, we rated the collaboration quality with a rating scheme that is based on the work of Meier et al. (2007; see also Rummel et al., 2011). Second, we compared the dyads’ eye gaze during various moments of the interaction to identify whether gaze indicators are sensitive to contrasts in the rating dimensions.

Rating Analysis

As the main goal of the current study was to identify simple gaze-based correlates of collaboration quality, we concentrated the analysis on two aspects of collaboration quality: communication and coordination (cf. Rummel et al., 2011). Two other aspects, joint information processing, as well as interpersonal relationship and motivation were not taken into account as these may have less clear indicators in dyadic gaze. We evaluated five dimensions that we briefly describe hereafter. Two dimensions assess the quality of the dyadic communication, and three dimensions assess the dyad’s coordination of joint work.

Collaboration flow evaluates if the collaboration partners engage in a coherent exchange of information and keep mutual awareness on what the partner is currently working on. High collaboration flow is marked by a seamless “flow” of dialogue while collaboratively solving the joint task, in other words, the dyad partners address each other and react to each other’s proposals or questions. The dimension *sustaining mutual understanding* evaluates if the dyad partners successfully work towards a shared basis of understanding, in other words, a common ground (cf. Clark & Brennan, 1991). Dyads that show successful grounding give and ask for feedback. Mutual understanding can also be facilitated if the dyad partners clarify about which aspect they will talk next, for instance by verbal references to the specific part of the code, or by selecting the specific part of the code in the shared workspace (deictic reference). The dimension *technical coordination* assesses if the dyad partners effectively use the technical environment during problem solving and capitalize on the system resources (cf. Diziol et al., 2008). In the current study, the dyad partners work on a joint work space: if one partner scrolls, the window of the other dyad partner changes as well; if one partner highlights code, his or her partner can see it. In contrast to Meier et al. (2007), the current study evaluates the coordination of the joint problem-solving with two separate dimensions. Similar to Meier et al. (2007), we evaluate the amount of *task division*, in other words: Do the dyad partners solve all aspects of the task together, or do they alternate between individual and collaborative work phases? In addition, the dimension *participation symmetry* assesses if both dyad partners are equally engaged in problem solving.

As in Meier et al. (2007), we assessed the dimensions on a 5-point rating scale from - 2 (low quality) to + 2 (high quality). The dimension task division was an exception. While phases of individual work are crucial for a successful outcome in *interdisciplinary* interaction (Rummel & Spada, 2005), it is not yet clear if a certain amount of task division is also beneficial in the current *pair-programming* setting, or if it is better if the dyad partners constantly interact with each other. We therefore evaluated the amount of task division with a rating scale from 0 (no task division) to 4 (high amount of task division). The two first authors of this paper separately rated the quality of the collaboration and reached a high agreement; disagreements in the ratings were solved by mutual discussion. Because of the small number of dyads, we cannot report kappa statistics about inter-rater agreement.

Cross-recurrence Plots

Cross-recurrence plots represent the time-dependence of two processes in a dynamical system. Because the graphical interpretation of cross-recurrence is not straightforward, we hereafter describe a schematic cross-recurrence plot along a real plot for illustration purpose (see Figure 1).

The horizontal axis represents time for the first collaborator and the vertical axis represents time for the second collaborator. Each pixel of the plot (Figure 1, right) corresponds to 200 milliseconds time slice (the duration of short gaze fixations are around 100 ms). For a pixel to be colored, the distance between the fixations of the two collaborators has to be lower than a given threshold (70 pixels in our case). Moments of recurrence appear colored in the plot (neither light gray nor white). Cross-recurrence plots show exclusively recurrent gaze. If two collaborators continuously looked at two different spots on the screen for the whole interaction, the resulting cross-recurrence plot would be completely blank (light gray in Figure 1, right). On the contrary, if the two collaborators looked at the same spot on the screen totally synchronously, the plot would show only a dark line on the diagonal. Pixels exactly on the diagonal of the plot correspond to synchronous recurrence, e.g. collaborators look at the "same" target at exactly the same time. Pixels below the diagonal show a lag for the first collaborator (the second collaborator looked at the target before) whereas the pixels above the diagonal show a lag for the second collaborator. Asymmetries above and below the diagonal line could therefore be indicative of leading and following behaviors.

In some cases, the colored areas on the real plot form rectangular shapes, which are represented by rectangles in the schematic plot (Figure 1, left). Labels refer to areas of interest on the screen (e.g. for a computer program *a* would typically refer to the method signature, *b* to the body of the method, and *c* to the return statement). On the horizontal axis, the labels of the arrows represent the areas of the stimulus that were looked at by the first collaborator. The corresponding sequence of areas of interest that were looked at is "*a->b->a->c->a*". The vertical axis represents the sequence of areas focused on by the second collaborator: "*a->c->b->a->c*". Following the diagonal, we move through the interaction, and we see that the dyad has looked together synchronously at the areas "*a->b->a->c*". Off-diagonal regions indicate asynchronous cross-recurrence, which happens when collaborators peek at or review places they have been or will be looking at together. For example, in the schematic plot (Figure 1, left), the area labeled *c'* indicates that the second collaborator has looked at *c* long before the dyad focuses on *c* together. The areas marked *a'* are indicative of the first collaborator looking back at area *a* after having looked at it together with the second collaborator on two occasions.

Hence, in the domain of program understanding, cross-recurrence plots not only represent the "togetherness" of collaborative code exploration, but also show whether the exploration of the program is rather linear (there is almost no off-diagonal activity because regions of code are looked at only once) or iterative (there are marked off diagonal regions that result from the collaborators repeatedly looking at the same regions of the code).

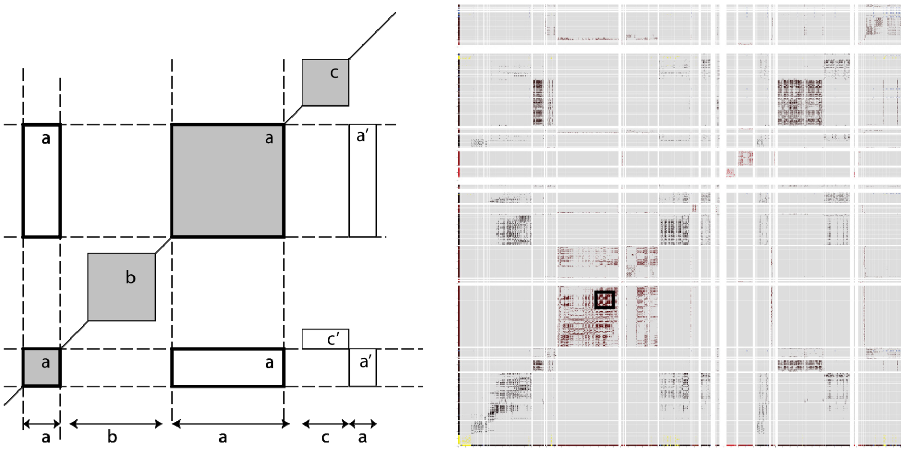


Figure 1. Schematic (left) and Real (right) Cross-recurrence Plots.

Results

Two dyads were chosen among available data at the time of writing because they best illustrated what is commonly understood in CSCL as a "good" and "bad" interaction quality. For each dimension of interaction quality (as defined by the rating scheme) we report a characterization of the interaction through gaze cross-recurrence plots and associated measures.

Gaze Correlates of the Dimensions of Interaction Quality

Overall, the collaboration quality of Dyad 1 is very bad. Their interaction is rather lopsided: Red explains to Blue how he understands the code, while Blue is hardly contributing to the discussion and back-channels only infrequently. For Red, it apparently was quite easy to understand the general functionality of the code, and except for a small mistake - he believes that the players can choose from a list of ten numbers instead of nine - he correctly explains the rules of the game. In contrast, Dyad 2 overall shows a high collaboration quality. After collaborators gained a good overview of the code, they jump to specific sections of the code in order to answer open questions. The dyad is able to explain the general functionality of the game.

In the remainder of this section, we describe the dyads' collaboration in more detail with reference to the dimensions of the rating scheme, and intertwine the descriptions with results from gaze analyses. Table 1 summarizes the ratings of the dyads' collaboration quality. Our goal in this contribution is to identify whether strong contrasts in ratings are reflected in behavioral measures.

Table 1: Collaboration quality of the two contrasting cases.

	Collaboration Flow	Mutual Understanding	Technical Coordination	Task Division	Particiation Symmetry
Dyad 1	-2	-2	-2	4	-2
Dyad 2	2	2	2	1	2

Collaboration Flow

The collaboration flow of **Dyad 1** is very bad. Mostly Red is talking, and he hardly reacts to Blue's attempts to interrupt his monologue. Only infrequently, Red addresses his partner to ask for his opinion. The unequal collaboration seems to frustrate Blue, and apparently, he capitulates: When recording the answer, Blue is no longer providing any back-channeling while Red engages in a monologue of almost four minutes. In contrast, **Dyad 2** shows a high collaboration flow. Only during reading the instructions, there is a short period where none of the subjects talk; but then they engage in a mutual discussion for the rest of the problem solving, and react to each other's questions and proposals.

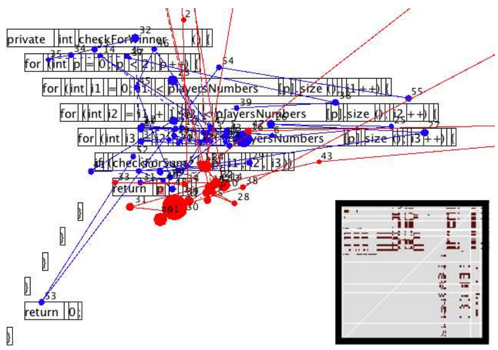
The two cross-recurrence plots that correspond to the 10 minutes of interaction are strikingly different (see Figure 2). For Dyad 1, the density of recurrence points on the diagonal is not very marked. The numerous white stripes are indicative of frequent scrolling. For Dyad 2, clearly defined rectangular areas appear along the diagonal of the plot. This pattern is indicative of gaze coupling, Red and Blue look at the same parts of the code within a few seconds of each other, and hence explore the code together, one criteria for a good collaboration flow. The square patterns are also indicative of some level of stationarity in the exploration of the code (maybe also related to the difficulty of the task), i.e. if the collaborators were constantly changing their focus, only a thin line would be visible on the diagonal. Off-diagonal rectangular patterns show that the dyad iterates through the code by revisiting sections of the code on several occasions.

Figure 2. Gaze Cross-recurrence Plots for Dyad1 and Dyad2. The bold rectangle delimits the moment where the dyads speak about the "check for winner" method and is analyzed below under "mutual understanding" (Full resolution images are available at http://bit.ly/9KQslc and http://bit.ly/aQbhk4).

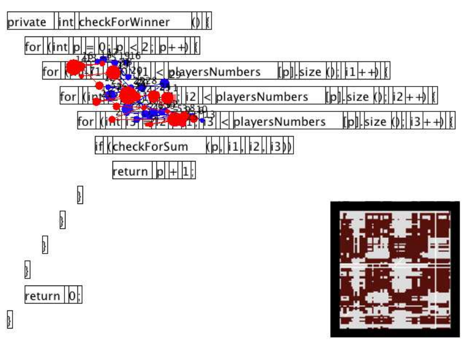
Sustaining Mutual Understanding

Dyad 1 displays difficulties in gaining a common ground. Although Blue sometimes uses verbal references to point out which part of the code he is looking at, the lack of back channeling from Blue indicates that he has difficulties in understanding his partner's comments. In contrast**, Dyad 2** successfully builds a common ground by frequently back-channeling, answering with relevant next turns, and providing verbal or deictic references to draw the partner's attention to specific sections of the code. For instance, towards the beginning of their collaboration, Red suggests, "shall we check the check for winner method?" and thereby guides his partner's attention. Similarly, during their collaboration, Blue highlights a line of the code to ask his partner to explain him the function of the line. However, the relation between selection and mutual understanding is not straightforward. From the dyadic data that we are currently analyzing on 42 dyads we see that there is no correlation between the proportion of interaction time during which subjects select some part of the code and the cross-recurrence. Selection alone is not favoring higher cross-recurrence. Indeed, in Dyad 1, at moments Red frantically selects text in pace with his own reading and speaking activity but without the intention to ground the focus of conversation with his partner.

To illustrate the correspondence between cross-recurrence plots, gaze patterns and dialogue, we compared the interaction of both dyads while they analyze the method "check for winner" in Figure 3. Red who is assisted by Blue only for a vocabulary question dominates the interaction for Dyad 1. While Red selects elements quite often while speaking, Blue does not follow his gaze. In Dyad 2, the two collaborators traverse the code synchronously and produce a socially distributed production started by Red and completed by Blue (Roschelle & Teasley, 1995).



Dyad 1: 56:40 to 57:17
Red: and this part, "check for winner", it actually, p, eh, first look p is for player and those indexes are, eh, for their, their list of taken numbers, and at one moment it simply … eh … gives those indexes and gives what eh what player eh
Blue: are chosen, yes,
Red: yeah thanks, and if that sum is equal to 15 it will simply return p + 1
Blue: hm hm



Dyad 2: 17:49 to 18:05
Red: oh look the initial, the start <select int before i1> of the loop <select int before i2>is different each time <select int before i3>
Red: here
Red: first I1 <select i1> starts at 0 <select 0>
Red: i2 starts at ...
Blue: yeah i2 will start a 1 ... and i3 will start a 3
Red: <select 1 after i2>
Red: yeah so we take the first second and 3rd number and we iterate like that-

Figure 3. Fixation Plot with the Corresponding Cross-recurrence Plot Detail (left) and Transcript for Explanations Given by Dyad1 and Dyad2. Both explanations are about the "check for winner" method.

Technical Coordination

Dyad 1 has a hard time to effectively coordinate the usage of the technical environment. Red is frequently scrolling up and down. The scrolling obviously hampers Blue in his attempts to understand the code: He frequently tries to counter steer by scrolling in the other direction and even explicitly addresses his difficulties "It's just hard to navigate into the program". Still, the difficulties remain, and it is not clear if Red realizes that the screen is shared and scrolling up and down affects both partners. **Dyad 2** successfully coordinates the technical environment. Already at the start of the collaboration, Red addresses the technical coordination by asking his partner if the scrolling also affects his screen. Also in the further progress of the collaboration, they frequently explicitly coordinate the scrolling; for instance, Blue asks his partner if he agrees to scroll further down. Furthermore, the dyad uses the highlighting function to guide their partner's attention. Symmetric white stripes in the cross-recurrence plots indicate scrolling. In Dyad 1, the scrolling stripes are much more dense than in Dyad 2. An indication for effective technical coordination is that gaze is recurrent soon after a scrolling episode, indicating that collaborators did not loose their partners when changing the focus. Because this is not very apparent on the plots, an associated numerical indicator would probably be more informative.

Task Division and Participation Symmetry

The interaction of **Dyad 1** is characterized by a high amount of individual work phases and task division. During the interaction, there are several sequences about a minute and longer where they collaborators do not engage in interaction. The collaboration of **Dyad 2** looks fairly different. There is only a short individual work period at the start of their problem solving where each collaborator reads the instructions; then they engage in mutual interaction. Task division results in low recurrence if it implies that collaborators look at different places of the workspace. Recurrence plots would feature a sequence of off-diagonal elements followed by diagonal recurrence at the end of the interaction if parallel work ended by sharing and reviewing the results. With **Dyad 1** however, the low recurrence seems to stem more from asymmetrical interaction rather than coordinated division of labor. Red basically answers the question on his own; even though Blue first tries to engage in the collaboration, he seems to give up during the further course of the interaction. In contrast, both partners of the **Dyad 2** are equally engaged in the participation, even though with slightly different roles: While Blue frequently asks questions for further clarification, Red provides more explanations. The different roles may be explained by the differences in the partner's prior knowledge.

Gaze recurrence is closely linked with effective dialogue. In **Dyad 1**, sustained audio signal only appears at minute 3 and is afterwards clearly asymmetric in favor of Red. Blue joins in for two minutes between 12 and 14 but then remains silent for the rest of the interaction. Blue only speaks for 4% of the time, whereas Red speaks for 15% of the total time. In **Dyad 2**, the Blue and Red collaborators hold the floor for respectively 41% and 43% of the total time. Of course, the asymmetry of the audio signal is much easier to measure than the level of gaze recurrence and is certainly a simpler indicator for participation asymmetry.

Cross-recurrence in Quantitative Approaches

We have seen the descriptive value of cross-recurrence plots by contrasting two examples of "good" and "bad" interaction quality. In more quantitative approaches to interaction analysis, researchers need a numerical counterpart to the visual patterns that represent the coupling of the collaborators gaze. High recurrence is visible on a cross-recurrence plot where points on the diagonal are dark. More, we are not only interested by the main diagonal (collaborators look at the same target exactly at the same time) but by an "interaction span" during which subjects' gaze follows their partners'. The graph in Figure 4 shows the rate of cross-recurrence as a function of time shifts between the fixations of one subject and the other. Each value on the x-axis corresponds to the time span between the main diagonal on the recurrence plot and a parallel diagonal shifted in time. The values on the y-axis represent the proportion of dark pixels on this diagonal. The cross recurrence is highest in the period of -2 to +2 seconds around the main diagonal, and we used the average value of the cross recurrence rate during this time span as a numerical indicator. The rate could be computed for longer periods of time (e.g. +- 4 and +- 6 seconds). When computed on a larger sample, the resulting indicators are highly correlated with the cross-recurrence rate for a span of +- 2 seconds. Cross-recurrence at much higher time shifts might be indicative of the collaborators revisiting of shared areas of interest (see our discussion above about off-diagonal regions in Figure 1) but does not indicate close coupling anymore.

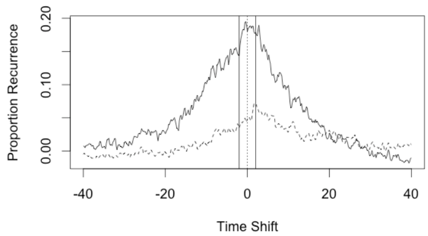


Figure 4. Cross-Recurrence Proportions by Time shifts for Dyad 1 (dashed line) and Dyad 2 (full line).

Conclusion

Contrasting two dyads provided first insights into how gaze cross-recurrence is related to dimension of the quality of collaboration. The most promising result is that high gaze recurrence seems to be typical of a "good" dyad where the flow of interaction is smooth and where partners sustain each other's understanding. It is now necessary to qualify the correspondence between the structure of interaction and graphical patterns in recurrence plots more systematically. One way to do this consists of computing numerical features that reflect graphical structure of the plots (e.g. symmetry, density, gaze transition paths from targets to targets) and to use them to discriminate between categories of interaction quality. An important question for these indicators is also whether they are task-specific and whether they depend on the stimulus displayed on the screen. Normalization of the recurrence measure with regards to a random baseline is required to be able to generalize the approach.

The use of low-level indicators that can be computed in real time would allow to a) automate the collection of evidence to provide real-time feedback, b) to inform and enrich the analysis of interaction by researchers.

Bibliography

Cherubini, M., Nüssli, M.-A., & Dillenbourg, P. (2008) Deixis and gaze in collaborative work at a distance (over a shared map): a computational model to detect misunderstandings. In Proceedings of the International Symposium on Eye Tracking Research & Applications (ETRA2008), p. 173-180. ACM Press, 2008.

Cherubini, M., Nüssli, M-A., & Dillenbourg, P. (2010). Indicating and looking in collaborative work at distance. Journal of Eye Movement Research 3(5):3, 1–205.

Clark, H. H., & Brennan, S. A. (1991). Grounding in Communication. In Resnik, L. B., Levine, J. M. & Teasley, S. D. (Eds.). *Perspectives on socially shared Cognition.* Washington: APA Books.

Dale, R., & Spivey, M. J. (2006). Categorical recurrence analysis of child language. *Language Learning.*

Dale, R., Warlaumont, A. S., & Richardson, D. C. (in press). Nominal cross recurrence as a generalized lag sequential analysis for behavioral streams. *International Journal of Bifurcation and Chaos*.

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers and Education, 46,* 6–28.

Diziol, D., Rummel, N., Kahrimanis, G., Guevara, T., Holz, J., Spada, H., & Fiotakis, G. (2008). Using contrasting cases to better understand the relationship between students' interactions and their learning outcome. In G. Kanselaar, V. Jonker, P.A. Kirschner, & F. Prins (Eds.), *International perspectives of the learning sciences: Cre8ing a learning world. Proceedings of the 8th International Conference of the Learning Sciences (ICLS 2008), Vol 3* (pp. 348-349). ISLS

Eckmann, J.-P., Kamphorst, S.O., & Ruelle, D. (1987) Europhys. Lett. 5, 973.

Firestone, W. A. (1993). Alternative arguments for generalizing from data as applied to qualitative researh. *Educational Researcher, 22*(4), 16-23.

Lord, R. G., & Levy, P. E. (1994). Moving from Cognition to Action: A Control Theory Perspective. *Applied Psychology: An International Review*, *43*(3), 335-398.

Meier, A., Spada, H. & Rummel, N. (2007). A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning, 2*, 63-86.

Nüssli, M.-A., Jermann, P., Sangin, M., & Dillenbourg, P. (2009) Collaboration and abstract representations: towards predictive models based on raw speech and eye-tracking data. In *CSCL '09: Proceedings of the 2009 conference on Computer support for collaborative learning*. ISLS, 2009.

Pietinen, S., Bednarik, R., Glotova, T., Tenhunen, V., & Tukiainen, M. (2008) 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 (ETRA '08)* (pp. 39-42): ACM, New York, NY, USA.

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*, 1045–1060.

Richardson, D. C., Dale, R., and Kirkham, N. Z. (2007). The art of conversation is coordination: Common ground and the coupling of eye movements during dialogue.*Psychological Science*, 18(5):407-413.

Roschelle, J., & Teasley S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (Ed.), *Computer-supported collaborative learning*. (pp. 69-197). Berlin: Springer-Verlag

Rummel, N., Deiglmayr, A., Spada, H., & Kahrimanis, G. (2011). Analyzing collaborative interactions across domains and settings: an adaptable rating scheme. In: S.Puntambekar, C. Hmelo-Silver, & G. Erkens (Eds.), *Analyzing interactions in CSCL: Methodologies, approaches and issues*. (pp. 367-390) New York: Springer.

Rummel, N., & Spada, H. (2005). Learning to collaborate: An instructional approach to promoting collaborative problem-solving in computer-mediated settings. *Journal of the Learning Sciences*, *14*(2), 201-241.

Soller, A., Martinez, A., Jermann, P., & Mühlenbrock, M. (2005). From mirroring to guiding: A review of the state of art technology for supporting collaborative learning. International Journal of Artificial Intelligence in Education, 15, 261–290.

Strijbos, J.-W., Martens, R. L., Prins, F. J., & Jochems, W. M. G. (2006). Content analysis: What are they talking about? *Computers & Education, 46*, 29-48.

Suthers, D., Law, N., Lund, K., Rose, C., & Teplovs, C. (2009). Common objects for productive multivocality in analysis. In C. O'Malley, D. Suthers, P. Reimann, & A. Dimitracopoulou (Eds.), *Computer Supported Collaborative Learning Practices - CSCL 2009 Conference Proceedings*, (pp. 225-226). International Society of the Learning Sciences.

Acknowledgements

This research was funded by the Swiss National Science Foundation (grants #K-12K1-117909 and #PZ00P_126611) as well as by a STELLAR incubator grant to the second author, Dejana Mullins.