Predicting Problem-Solving Behavior and Performance Levels from Visual Attention Data

Shahram Eivazi
School of Computing
University of Eastern Finland
seivazi@cs.joensuu.fi

Roman Bednarik
School of Computing
University of Eastern Finland
bednarik@cs.joensuu.fi

ABSTRACT
Inferring high-level cognitive states during interaction with a user interfaces is a fundamental task in building proactive intelligent systems that would allow effective offloading of mental operations to a computational architectures. In this paper, we propose a system that uses real-time eye-tracking to measure user's visual attention patterns and infers behavior during interaction with a problem solving interface. Using a combination of machine learning techniques, computational modeling and eye tracking, we investigate 1) the differences among good and poor performance groups, and 2) the possibility of inferring distinct cognitive states such as planning or cognition. We employ and train a support-vector machine (SVM) to perform a classification task on a set of features computed from eye movement data that are linked to concurrent high-level behavioral codes based on think aloud protocols. We contend that differences among cognitive states can be inferred from overt visual attention patterns with accuracy highly over chance levels. We observe that such a system can also classify and predict performance with up to 87% recognition rate for an unseen data vector for two classes. We suggest a prediction model as a universal model for understanding of complex strategic behavior. The findings confirm that eye movement data carry important information about problem solving processes and that proactive systems can benefit from real-time monitoring of visual attention.

Author Keywords
Eye-tracking, machine learning, proactive intelligent systems

ACM Classification Keywords
H1.2. Models and principles: User/machine systems.

INTRODUCTION
Modeling human behavior is one of the main challenges to create new adaptive interfaces that can understand user behaviors based on relevant user information record. The traditional data collection methods for the modeling task, such as logs or verbal data, are often not completely reliable or applicable. For instance, it has been frequently argued that tasks such as reading, mental computations, and problem solving are difficult to be assessed by traditional methods such as verbal protocol [9].

Eye tracking is considered as a technology that provides an unobtrusive, sensitive, and real time behavioral index of ongoing visual and cognitive processes. New, reliable and more comfortable eye trackers have become available. The availability of new eye-trackers motivates HCI researchers to employ them as input devices in real time interfaces [10]. For example, the technology has been applied in eye-typing [17], object pointing and selection [21], gaming [23], or interaction with problem solving [2].

Previous research also shows that eye movements during the observation of complex visual stimuli are regular and systematic (e.g. Yarbus [27] and Rayner [20]) which gives a motivation for modeling cognitive and perceptual processes based on eye-movement data. For example, differences between skilled and novice users have frequently been linked to the differences in the eye-movement patterns.

Modern eye-tracking research tends to rest on the eye-mind hypothesis [11]; eye-tracking data are commonly considered as a measure of overt visual attention and that is linked to the internal processing. Analysis of the relations between eye movements and human cognition has indeed proven fruitful in many domains, such as reading comprehension, visual search, selective attention, and studies of visual working memory [13]. Loboda and Brusilovsky [16] and Bednarik [3] argued that eye tracking can be applied in the area of user modeling and adaptive tools for improving the accuracy of prediction models. Loboda and Brusilovsky pointed to the advantages of eye movement data for on-line assessment of user meta-cognitive behavior. Conati and Mertena [4] showed that eye-tracking data improves the performance of probabilistic models in online assessment.

In this paper we describe the design and components of a system that employs eye-tracking data to model user performance and cognitive activities in an interactive problem solving task. The system consists of two prediction

2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction.
February 13, 2011, Palo Alto, California, USA.

models to provide a comprehensive recognition and unambiguous interpretation of eye gaze pattern in order to feed new intelligent user interfaces with behavioral predictions.

RELATED WORK

People apply a range of different strategies when they have to make a choice or decision to achieve their goals. Understanding these processes as they occur with interactive interfaces is not an easy task, but at the same time, it is a central research problem. Understanding user's plans and goals in real time would enable us to significantly improve interactive systems. Therefore, in order to create interfaces that are more sensitive to user's needs, the user's cognitive states must first be invariably recognized.

Ericsson and Simon [6] supported the idea of applying think aloud data to understand cognitive processes. They assumed that think aloud reports are a reflection of the cognitive processes that generate user's behavior and action. So far it is not however clear whether we can model aspects of the human mind only with verbal protocol. In real-time systems data collection with verbal protocols methods is problematic, because think aloud utterances are often incoherent [6] and verbalizing thought is not natural in everyday situations. Van Someren et al. [25] argued that in many cases it is possible to combine think aloud method with other data collection methods. Think aloud method is used to report data. Later this data can be used to support and promote analysis of other methods.

Another data collection tool frequently applied to get insights into cognition is eye tracking. Analyses based on eye-tracking have several advantages over other protocols. Glöckner & Herbold [7] argued that in a problem solving experiment recorded data from users with eye tracking methods decrease the chance of influence on decision process.

Goldberg and Kotval [8] argued that eye tracking is one of the particularly strong methods in the assessment of users' strategies. They consider eye movement-based analysis as an evaluation technique that enhances the traditional performance data such as think-aloud protocols, and walk-through evaluations of computer interfaces.

With few notable exceptions (e.g. Anderson et al. [1]) it is generally accepted that eye movements, eye-fixations and the derived measures provide information about cognitive processes. For instance, Velichkovsky et al. [26] claimed that fixation durations increase during solving a problem with increasing the level of cognitive processing. Thus short fixations are related to more superficial levels of processing (e.g. screening or perception), whereas longer fixations are related to deeper processing (e.g. deliberate consideration of information and planning) [7].

Recent empirical data obtained from eye movement models (e.g. EMMA Salvucci [21]) provide a good motivation for building systems that can adapt to users interaction with the environment and learn from eye movement data.

Both user expertise and cognitive states have been previously modeled using eye-tracking data. Based on machine learning classification Liu et al. [15] explained the differences between experts and novices in building concept maps. Participants constructed collaboratively concept-maps of the content in the text for 20 minutes as their eye-movement data were recorded. Results showed 96% recognition rate for two distinct clusters (experts and novices). The authors reported that while higher-skilled participants concentrated on specific concepts longer, lower-skill participants had shorter attention spans and scattered gazes. In another experiment Liang et al. [14] claimed that a general Support Vector Machine (SVM) is a strong machine learning method for classification of human behavior, especially for detecting cognitive states via eye movement data. Authors demonstrate that driver distraction can be detected using driver performance measures and eye movement measures in real time.

In another study, Simola et al. [22] applied Hidden Markov Models to predict what task a user is currently conducting out of three information search tasks: word search, searching for an answer, or choosing most interesting title from a list. The model was trained on eye-tracking data and achieved an accuracy of 60.2%.

MAPPING GAZE DATA TO SEMANTIC CONCEPTS

Modeling internal cognitive states is an active research topic, however complex problem solving is a domain not previously explored in greater detail using eye-movement tracking. Yet, in the domain of intelligent user interfaces in order to support the user's interaction with a system, the IUIs have to accurately tap into the sequence of thoughts of people.

In this study we thus employ eye tracking to reveal such relevant information from user's ocular behavior. Gaze data are associated with human cognition states by using think aloud protocol as a ground truth. The presented method and system includes the following main parts: verbal protocol analysis of the cognition, feature extraction and mapping to the verbal protocols, and machine learning method for building associations between the two. First, we code all think-aloud data by listening to the user's speech during interaction. We applied a coding scheme based on O'Hara & Payne [19] method that is based on Ericsson's approach [12] and has also been applied with modifications in other studies (e.g. Morgan et al. [18]). In the second phase, we propose a novel way of mapping gaze-based data to qualitative differences in the corresponding think-aloud protocols. We compute a set of eye-tracking features that are informed by the theories of cognition and visual attention and for each data-point in the think-aloud protocol we build a corresponding vector of these features. In the last stage, we present the inference task as a typical classification problem and we apply machine learning and

pattern recognition methods to solve it. Figure 1 presents the computational architecture of the proposed approach.

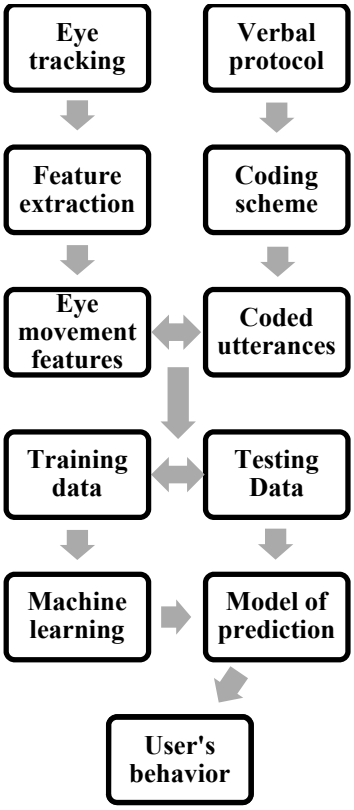

Figure 1. Procedures of the proposed mapping.

The mapping system described above enables us to 1) investigate the relationships between high-level cognitive traits and low-level eye-tracking data, and 2) propose a prediction real-time model to recognize user's cognitive states and user's performance. Future interactive systems can make use of the automatic modeling and classification methods proposed in this paper.

In the rest of this paper we conduct a feasibility evaluation of this approach in the domain of interactive problem-solving. We propose the feature set and we evaluate the accuracy of the approach.

EXPERIMENTAL DESIGN AND PROCEDURE

In order to answer the question whether gaze data can be used to classify and predict human strategies and performance we choose the classical 8-tiles puzzle game. We employ the data collected from the experiment of Bednarik et al. [2]. Similar settings have been used in numerous studies investigating interactive human problem solving. The authors had instructed a group of participants to think aloud while solving the 8-tiles puzzle game. Each tile in the puzzle had dimensions of 200×200 pixel (for each tile: width was 5.29 cm and height was 5.29 cm, measured on the screen).

Fourteen participants solved three trials of the game using computer mouse as an interaction method. They started with a warm-up puzzle and a think aloud practice and then continued for three unique initial configurations of the puzzle game. The three configurations were comparable in the level of complexity and were presented in random order. The target desired state of the puzzle is shown in Figure 2. Figure 3 present the three initial states of the puzzle game.

In addition to participants' voice protocols, eye movements were recorded using Tobii ET 1750 eye tracker. The resolution of the 17 inches screen was 1280×1024 and the viewing distance 60 cm [2].

Data from two participants have been removed because of low quality of eye tracking data. Preliminary eye movement data analysis has been performed with Clearview version 2.7.1 (http://tobii.se), with a default setting for fixation identification algorithms. MATLAB version R2009b and LibSVM Matlab toolbox of [10] have been used for the data analysis.

DATA ANALYSIS

To achieve the goal of predicting user characteristics and skills through the eye movement data, two main analysis techniques had to be carried out. First, outcome measures have to be defined and computed, including feature extraction and clustering of the features. Second task consisted of creation and validation of the prediction model.

Outcome measures

To address the first problem (outcome measures), verbal data were classified into six categories based on O'Hara & Payne [19] with a slight modification. The classification categories described qualitatively different utterances: *Cognitions* referred to statements describing what concrete and specific information a participant is currently attending to. *Evaluations* were conceptually similar to cognitions while, they were less accurate about the object of interest. In addition, when participants were referring to how well

Figure 2. Goal state of 8- tiles puzzle game.

Figure 3. Three initial states for 8- tiles puzzle game.

they performed or what is the general situation in the problem-space, we coded that utterance as belonging to evaluations. *Plans and planning* were utterances containing a plan development, its specific goals and detailed actions to be taken next. *Intentions*, on the other hand, were utterances describing the general aims, without a specific descriptions how to achieve them. *Concurrent move* utterances referred to description of the changes in the problem along the manipulation with it. Finally, we applied a category of *not applicable* for other utterance; however, we do not consider those data in this analysis. More detailed description can be found in [19].

The unit of analysis was one sentence. Two independent coders conducted the coding and achieved the inter-rater agreement of 86%. Since all classified utterances included a time stamp, the action times spent on utterances were taken into an account as feature in this study.

Of all three trials and all participants, the coding yielded a total of 249 data points for Cognition states, 339 data points for Evaluation activities, 105 data points for Planning, 235 data points for Intention related utterances, and 318 utterances containing the descriptions of concurrent moves.

In this study the outcome features of eye movement data were based on fixation duration and the location of the eye movements with respect to the screen coordinates. The eye movement features that used in this experiment are listed in Table 1. Similarly as the coded utterances, eye-movement data carried a timestamp, enabling their easy mapping with the verbal protocol. In our case, each feature vector was computed from eye-movement data for the duration of the corresponding utterance. For example, during three seconds of one coded state, such as 'evaluation', we computed all eye movement features from this interval.

Furthermore, we partitioned the screen into areas of interest (AOIs). The user interface was partitioned into nine AOIs corresponding with the nine possible positions of tiles of the game, and one additional surrounding area for the remaining part of the screen. The goal state of the game was showed constantly at the left bottom of the screen.

Prediction model

To address the second task (prediction model) we employ a well-established machine learning approach. SVM is a standard and frequently applied tool that has been previously shown performing well for various classification tasks [17]. SVM has been successfully used in detection, verification, recognition, and information retrieval from a range of datasets [14]. Liang et al. [14] presented three arguments that make SVM suitable for classification of human cognition states: first, it is rarely possible to represent cognitive states of humans by a linear model. SVM can compute nonlinear models as efficiently as the linear models. Second, SVM can be applied without prior knowledge before training. In addition, it can extract information from noisy datasets. Third, while traditional learning methods (e.g., logistic regression) only minimize training error, SVM minimizes the upper bound of the generalization error. This makes SVM able to produce more robust models. In our application, SVM is used as a supervised learning classification method.

Eye movement feature	Description
Mean fixation duration	The average time of fixation duration in each state of coding scheme.
Sum of fixation duration	Sum of times of fixation durations in each state of coding scheme.
Mean path distance	The average distance of two consecutive fixations in each coding state based on eyes' coordinates
Total path distances	Summation distance of eye movements in each coding state based on eyes' coordinates
Number of fixations	Number of fixation during of each coding state
Fixation rate	Number of fixation divided by the duration of each coding state
Visited tiles (rate)	Number of visited tiles divided by number of fixation in each coding state

Table 1. List of eye movement features.

We built two prediction models. The first model learns the patterns of human cognitions (five states) and eye movement features. The second model searches for patterns between data vectors originating from different performance groups (two or three classes, high-, medium-, and low-performing participants) and eye movement features. The task is to predict, to which performance group any given data vector belongs.

For the former, the ground truth was labeled for each class (five coding states) in the sample data. For the latter, the ground truth was established by computing the task completion times. The data were split into training and testing datasets so that the data from two trials were considered as training, and the remaining trial as testing (unseen) data. Both training and testing data were normalizing between [0 1] with the same method as presented below:

$$\delta = \frac{d - d^{min}}{d^{max} - d^{min}} \ ,$$

where $\delta = normalized\ vector,\ d = original\ vector.$

Normalization of data was applied in two ways for the two prediction models. In the case of cognition recognition, we defined minimum and maximum values in training and testing dataset separately. In the case of performance recognition we defined the minimum and maximum values in the training and testing datasets for each participant individually.

We used Libsvm Matlab toolbox developed by Hsu et al. [10] to build the prediction models. In order to find best

parameter for the model, we employed a 3-fold cross validation method. Experimentally, we learned that for n>3 in n-fold cross validation, accuracy has not been changed significantly. Therefore, the training data has been divided into three subsets. Consequently, one subset was tested by using the model based on the remaining datasets (two subsets). At the end, cross validation accuracy was equal to the percentage of data that was correctly classified [10]. C-SVC support vector classification with RBF kernel has been applied.

RESULTS AND DISCUSSION

In this paper we analyze user behavior during a problem-solving task. In particular, we analyze the eye movement data and features as subsets aligned with the categories of verbal protocols. We first present the results related to the classification and inference of cognitive states alone, then we introduce the classification based on performance differences, and finally we present a combination of the two.

A complete description of the mean values and standard deviations of the features computed for each of the cognitive states can be found in [5][1]. The differences in individual features related to the cognitive states were generally small and the features contained great variances.

The baseline performance was established as a classification accuracy of a majority classifier. Given the fact that most of the classes belonged to the Evaluation class, the majority classifier would perform with accuracy of 27%. Table 2 shows the recognition accuracy of the SVM for the five cognitive states (cognition, evaluation, planning, intention and concurrent moves). On unseen samples the accuracy was about 53%. These results are reported also in [5] and we report them here for completeness.

The breakdown of the results indicates that cognition is the hardest activity to automatically recognize, as seen from the confusion matrix (see Table 3). By removing cognition-data from the dataset we were able to increase the recognition accuracy up to 64%.

In addition to the cognitive states prediction, we investigate how well any given data vector informs about the originating performance group. All users were divided into three groups; we denoted these groups as high-performance, medium-performance, and low-performance groups.

[1] Before the original paper will be published (ACM Press), we provide a reference to a table describing the data: http://cs.joensuu.fi/~seivazi/koli.JPG

	Cross validation	Unseen data
Accuracy	75.84	53.25
Penalty parameter of the error term in RBF kernel (C)	64	64
Parameter of RBF kernel	0.25	0.25

Table 2. Cognition state activities recognition.

Prediction outcome %						
Actual class		Cognition	Evaluation	Planning	Concurrent move	Intention
	Cognition	2.6	24.67	3.9	67.53	1.3
	Evaluation	1.9	96.19	0	0.95	0.95
	Planning	7.69	25.64	48.72	12.82	5.13
	Concurrent move	5.68	28.41	1.14	64.77	0
	Intention	15.79	43.42	0	6.58	34.21

Table 3. Confusion matrix.

In this analysis, the high-performance group contains four participants who solved the puzzle with average tasks completion time less than 120 seconds.

The medium-performance group contains five participants who solved the puzzle with average tasks completion time between 120 and 240 seconds. Finally, the low-performance group contains three participants who solved the puzzle with average tasks completion time more than 240 seconds. The pair wise differences in the average completion times between the groups were significant.

The features for each of the groups are compared in Table 4. It is worth to note that the action time on utterances for the high-performance group was much shorter than that of the low-performance group and that the standard deviation of the high-performance group was low.

Other observation relates to the fact that while the high-performance group had lower number of fixations, they had longer fixation durations. In other cases, however, it is hard to visually spot eventual patterns of differences between the groups, partly due to the great variances.

Table 5 presents the recognition accuracy of predicting into which of the three groups an arbitrary vector of data belongs. The accuracy of 66% can be considered relatively low, however the baseline classifier would achieve only 55% accuracy.

To test the influence of the data from the medium-performing group, we removed the data and conducted the classification again only for the two remaining groups. We speculated the medium-group dataset can contain such

Groups (Number of participants) Type of feature	High-performance (4)		Medium-performance (5)		Low-performance (3)	
	Mean	SD	Mean	SD	Mean	SD
Mean action time on utterance (ms)	4336.03	2943.31	7495.02	9026.77	7399.38	8866.92
Mean number of fixations average	10.85	7.97	18.14	21.91	16.08	17.56
Mean fixation duration average (ms)	266.64	114.85	288.52	107.52	206.67	58.91
Mean path-distance average	168.87	68.42	169.48	84.33	191.13	91.62
Mean fixation duration summation (ms)	2704.34	1887.78	4922.55	5896.40	3213.65	3367.36
Mean path-distance summation (pixels)	1819.94	1472.00	2739.43	3135.48	3130.68	3836.88
Mean fixation rate (Hz)	2.52	0. 69	2.43	0.60	2.54	0.83
Mean visited tiles rate	0.511	0.21	0.458	0.23	0.491	0.24

Table 4. Comparison of features among High, Medium, and Low-performance groups. SD= Standard deviation.

feature-spaces that could be overlapping with data either of the two other groups.

The accuracy of 87.5% (Table 6) shows that the chance of correct prediction whether a data point belong to either to a high or low-performance group is indeed high. In fact, we achieved 66.18% accuracy of correct recognition for data vectors from the high-performance group and 96.79% accuracy for the low-performance group. In other words, if

	Cross validation	Unseen data
Accuracy	80.82	66.48
Penalty parameter of the error term in RBF kernel (C)	256	256
Parameter of RBF kernel	1	1

Table 5. Three-group recognition rate (Low, Medium, or High-performance group).

	Cross validation	Unseen data
Accuracy	96.41	87.50
Penalty parameter of the error term in RBF kernel (C)	32	32
Parameter of RBF kernel	1	1

Table 6. Two-group recognition rate.

the classifier processed a data point from a low-performing user, in about 97% the data were correctly classified. The smaller size of the high-performance group dataset (fewer participants and faster completion time leading to less data for training) can be the main reason for lower recognition rate for the high-performance group. An improvement can be achieved by adding a weight in SVM parameters (w=1.5) to the high-performance group. In that case, the individual accuracy for high-performance group users can be increased to 73.53%, however, with a trade-off as the accuracy for the low-performance group users slightly decreased to 94.87%.

Finally, we conducted the classification task separately for the five cognitive states in each performance group, see Table 7. The results show that recognizing cognitive activities of high-performers is rather difficult; again, the reason can be found in the small sample size. On the other hand, in about 75% of cases of medium- and low-performing users the classification correctly predicted what cognitive activity a user is currently undertaking. While the recognition rates are still relatively low in absolute values, they are still high when compared to a baseline recognition rate.

Descriptive statistics of Table 7 is shown in the Figure 4 and Figure 5. The Figure 4 shows the mean path distance for each performance group. The chart presents the fact that path-distance shows a U-shape behavior in at least three cognitive states: planning, concurrent move, and intension states are characterized by similar means and variances in the high- and low-performing groups, while the medium-performance group shows a decrease in the measure. The U shape behavior repeats in the Figure 5 for the measure of the rate of visited tiles, however the variance in the rates of visited tiles is high.

CONCLUSION AND FUTURE WORK

Computing and HCI researchers rooted in cognitive-science tradition frequently assume that the mind consists of mental representations and structures comparable to computer data structures, and it executes computational procedures similar to computational algorithms [24]. While we do wish to remain neutral to these views, the results presented here let us to suggest that at least some sub-part of cognition and user traits can be modeled effectively using traditional computational principles and methods.

	Baseline	Cross validation	Unseen data
High-performance (4)	53	53.38	36.36
Medium-performance (5)	53	73.30	37.10
Low-performance (3)	42	75.34	47.40

Table 7. Accuracies of cognitive state recognition for five cognitive states among High, Medium, and Low-performance groups. Size of group in parenthesis.

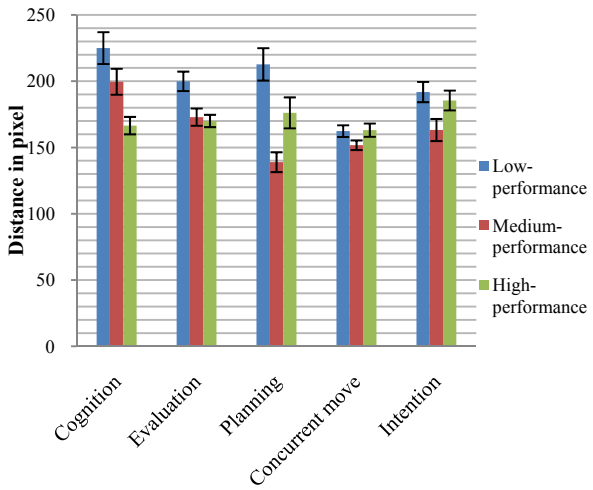

Figure 4. Mean path distance for each performance group.

We applied a SVM-based classification to predict, firstly, problem-solving cognition states and, secondly, user's performance. The goal was to evaluate, whether eye tracking can be used to detect cognitive behavioral patterns for a purpose of proactive intelligent user interfaces. We combined the approaches of machine learning methods, cognitive science, and HCI to describe a design and components of the real-time eye-tracking system for measuring user's visual attention patterns and inferring user's behavior during interaction with a problem-solving interface.

The novel result presented here shows that although the differences in problem-solving and related eye-movement data are subtle and multidimensional, they can be automatically recognized using SVM classification, with more than 87% accuracy.

This leads us to a conclusion that prediction of the user performance is possible, can be automated, and that the eye movement data carry important information about the skills of participants.

While the accuracy of classifications of cognitive activities was not extremely great, our finding shows that eye movement data carry important information about the problem solving process. We believe that increasing the sample-size to feed the training system can improve the accuracy. In addition, in the experiment we assumed that users' utterances always belong to whatever action they had just taken. The analysis of the verbal data showed that this was not always the case, particularly for high-performing experts. The expert participants often had to be prompted by the observers to talk, thus some of the thoughts were not captured in the protocol and some of the utterances that they shared were not aligned with the current eye-

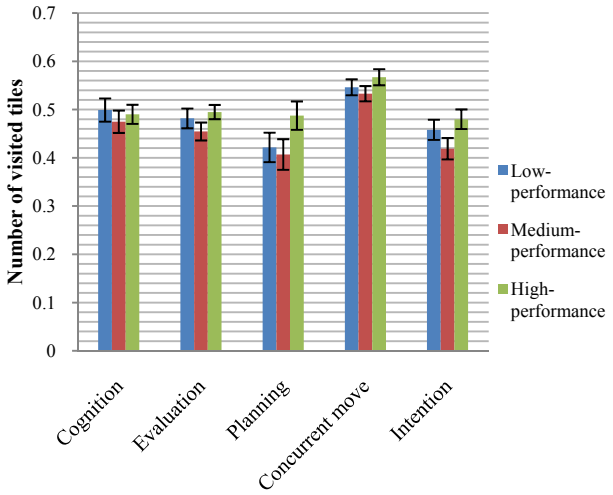

Figure 5. Mean rate of visited tiles for each performance group.

movement data simply because the style of the verbalization changed from concurrent thinking to retrospection. We plan to take this consideration into account, by both simplifying the coding protocol and extending the boundaries of the sample window to include the previous samples into mapping.

The future steps of this research include a development of a real-time system that dynamically captures and classifies user traits based on eye-movement data. In the domain of problem-solving this will enable us to build an intelligent environment that closely follows the user action and can proactively provide guidance, for example for the purposes of learning.

REFERENCES

1. ANDERSON, J. R., BOTHELL, D., AND DOUGLASS, S. 2004. Eye movements do not reflect retrieval: Limits of the eye-mind hypothesis. *Psychological Science 15*, 225–231.
2. BEDNARIK, R., GOWASES, T., AND TUKIAINEN, M. 2009. Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research*, 1–10.
3. BEDNARIK, R. 2005. Potentials of Eye-Movement Tracking in Adaptive Systems. In Proceedings of the

4th Workshop on the Evaluation of Adaptive Systems, held in conjunction with the 10th International Conference on User Modeling (UM'05), 1-8.

4. CONATI, C. AND MERTEN, C. 2007. Eye-tracking for user modeling in exploratory learning environments: an empirical evaluation. *Knowl. Based Syst.* 20, 557–574.

5. EIVAZI, S. AND BEDNARIK, R. 2010. Inferring problem solving strategies using eye-tracking: System description and evaluation. In *Proceedings of Koli Calling, International Conference on Computing Education Research, ACM press*.

6. ERICSSON, K. A. AND SIMON, H. A. 1993. *Protocol Analysis: Verbal Reports as Data Revised Edition*. MIT press.

7. GLÖCKNER, A. AND HERBOLD, A.-K. 2010. An eye-tracking study on information processing in risky decisions: Evidence for compensatory strategies based on automatic processes. *Journal of Behavioral Decision Making*.

8. GOLDBERG, J. H. AND KOTVAL, X. P. 1999. Computer interface evaluation using eye movements: methods and constructs. *Industrial Ergonomics 24*, 631–645.

9. SURAKKA, V., ILLI, M. AND ISOKOSKI, P. 2003. Voluntary eye movements in human-computer interaction, chapter 22, page 471. North Holland.

10. HSU, C.-W., CHANG, C.-C., AND LIN, C.-J. 2003. A practical guide to support vector classification. Tech. rep. Department of Computer Science and Information Engineering, National Taiwan University

11. JUST, M. A. AND CARPENTER, P. A. 1976. Eye fixations and cognitive processes. *Cognitive Psychology*, 441–480.

12. ERICSSON, K. A. 1975. Instruction to verbalize as a means to study problem-solving processes with the 8-puzzle. Tech. rep., Department of Psychology, University of Stockholm.

13. KALLER, C. P., RAHM, B., BOLKENIUS, K., AND UNTERRAINER, J. M. 2009. Eye movements and visuospatial problem solving: Identifying separable phases of complex cognition. *Psychophysiology 46*, 818–830.

14. LIANG, Y., REYES, M. L., AND LEE, J. D. 2007. Real-time detection of driver cognitive distraction using support vector machines. *Intelligent Transportation Systems, IEEE Transactions on 8*, 340 – 350.

15. LIU, Y., HSUEH, P.-Y., LAI, J., SANGIN, M., N¨USSLI, M.-A., AND DILLENBOURG, P. 2009. Who is the expert? analyzing gaze data to predict expertise level in collaborative applications. In *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, 898 –901.

16. LOBODA, T. D. AND BRUSILOVSKY, P. 2010. User-adaptive explanatory program visualization: evaluation and insights from eye movements. *User Modeling and User-Adapted Interaction 20*, 191–226.

17. MEYER, D., LEISCHA, F., AND HORNIKB, K. 2003. The support vector machine under test. *Neurocomputing 55*, 169–186.

18. MORGAN, P. L., WALDRON, S. M., KING, S. L., AND PATRICK, J. 2007. Harder to access, better performance? The effects of information access cost on strategy and performance. *In proceedings of the 2007 Conference on Human interface 4557*, 115–125.

19. O'HARA, K. P. AND PAYNE, S. J. 1998. The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology 35*, 34–70.

20. RAYNER, K. 1998. Eye movements in reading and information processing:20 years of research. *The American Psychological Association*, 372–422.

21. SALVUCCI, D. D. 2001. An integrated model of eye movements and visual encoding. *Journal of Cognitive Systems*, 201–220.

22. SIMOLA, J., SALOJÄRVI, J. AND KOJOC, I. 2008. Using hidden markov model to uncover processing states from eye movements in information search tasks. *Cognitive Systems Research* 9, 237–251.

23. SMITH, J. D. AND GRAHAM, T. C. N. 2006. Use of eye movements for video game control. In *ACM International Conference Proceeding Series*.

24. THAGARD, P. 2007. Cognitive science, Stanford encyclopedia of philosophy. http://stanford.library.usyd.edu.au/entries/cognitive-science/

25. VAN SOMEREN, M. W., BARNARD, Y. F., AND SANDBERG, J. A. 1994. *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press.

26. VELICHKOVSKY, B.M., ROTHERT, A., KOPF, M., DORNHOEFER, S.M. AND JOOS, M. 2002. Towards an express diagnostics for level of processing and hazard perception. *Transportation Research* 5, 145-156.

27. YARBUS, A. 1967. Eye movements and vision. *New York, Plenum Press*.