Dazed: Measuring the Cognitive Load of Solving Technical Interview Problems at the Whiteboard

Mahnaz Behroozi1, Alison Lui2, Ian Moore1, Denae Ford1, Chris Parnin1
1North Carolina State University, Raleigh, NC, USA
2University of Notre Dame, Notre Dame, IN, USA
{mbehroo,ipmoore,dford3,cjparnin}@ncsu.edu,alison.m.lui.2@nd.edu

ABSTRACT
Problem-solving on a whiteboard is a popular technical interview technique used in industry. However, several critics have raised concerns that whiteboard interviews can cause excessive stress and cognitive load on candidates, ultimately reinforcing bias in hiring practices. Unfortunately, many sensors used for measuring cognitive state are not robust to movement. In this paper, we describe an approach where we use a head-mounted eye-tracker and computer vision algorithms to collect robust metrics of cognitive state. To demonstrate the feasibility of the approach, we study two proposed interview settings: on the whiteboard and on paper with 11 participants. Our preliminary results suggest that the whiteboard setting pressures candidates into keeping shorter attention lengths and experiencing higher levels of cognitive load compared to solving the same problems on paper. For instance, we observed 60ms shorter fixation durations and 3x more regressions when solving problems on the whiteboard. Finally, we describe a vision for creating a more inclusive technical interview process through future studies of interventions that lower cognitive load and stress.

CCS CONCEPTS
• Software and its engineering → Software creation and management;

KEYWORDS
technical interviews, cognitive load, eyetracking

1 INTRODUCTION
A technical interview is a stage of a job interview, which for software developers, often includes a programming component performed on a whiteboard. A common goal of a technical interview is to obtain visibility into and verifiability of the cognitive processes used by a candidate [8]. Although technical interviews are the most common assessment technique, certain types of interviews can unnecessarily affect the candidate’s cognitive load, the total information demand placed on an individual [14]. There are several sources of cognitive load:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSE-NIER’18, May 27-June 3, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5662-6/18/05...$15.00
https://doi.org/10.1145/3183399.3183415
we obtained several measures consistent with stress. To investigate the feasibility of using this approach, we conducted a pilot study with 11 participants, where participants wear a head-mounted eye-tracker (see Figure 2) and we measured the difference between solving a programming task on paper and on the whiteboard. Using a Latin Square design to rotate experimental conditions, we observed that participants solving problems on the whiteboard appear to experience higher cognitive load. Further, participants self-rated the whiteboard setting as being more stressful, we observed several nervous tics displayed by participants, and we obtained several measures consistent with stress.

Performing an analysis of the different cognitive states of interview candidates can help define the challenges and benefits of technical interview styles and how different settings affects a candidate’s state of mind. As a result, we can use these models of attention to design better interview procedures that minimize disruption to candidates while allowing interviewers to assess a candidate’s thought process and problem solving speed.

2 BACKGROUND

Eye-tracking is the process of locating the eye-position and measuring the eye-movement of a subject. Eye-trackers are designed to monitor and collect eye-movement data while the subject is looking at a stimulus during an experiment [6, 18]. A stimulus is an object, such as a piece of code, which is of interest during an eye-tracking study. Generally, there are two types of eye-trackers: head-mounted eye-trackers and remote eye-trackers. Head-mounted eye-trackers have embedded infrared cameras and the subjects wear them. Remote eye-trackers have a screen with sensors and the subjects sit in front of them, such as the ones used to study eye movements of developers while using IDEs [20, 21].

Some common eye-tracking terms and measures are as follows:

- **Areas of Interest (AOI):** A fixed region corresponding to an object of study.
- **Fixation or Visual Intake (VI):** Stabilization of the eye on a particular point of the stimulus, typically between 200 to 300 ms. Fixations can be acquired from the eye-tracker in the form of a time stamp and x-y pixel coordinates. Privitera et al. suggest that most of the cognitive processes and information acquisitions happens during fixations [16].
- **Saccade:** A sudden rapid eye movement which occurs between two fixations, typically lasting between 40–50 ms. Prior studies claim that cognitive processing and information acquisition is not notable during saccades [6, 16].
- **Regression:** A backward movement of the eye from an AOI to a previously visited AOI.
- **Cognitive Load:** Several studies report eye-tracking measures which reveal cognitive load. Chen et al. conclude that fixation duration and fixation rate are indicators of an increment in the attention [4]. Increase or decrease of fixation duration depends on the characteristics of the task. Fixation duration will decrease in stressful tasks that need rapid responses (such as driving in a car [23] or airplane piloting [10]). In contrast, in the tasks that needs more cognitive processes such as reading texts of increasing complexity, fixation duration will increase [17]. Chen et al. [4] also included the measurements of saccade velocity and saccade length in order to investigate human mental effort. Their results show that saccade velocity and length are highly discriminatory parameters. Similarly, Manuel et al. [12] find that a decrease in saccade velocity indicates tiredness and an increase of saccade velocity indicates a higher task complexity. High blink latency and a low blink rate indicate high mental effort [4].

3 APPROACH

3.1 Placing boundaries for the coding area

The SMI head-mounted eye-tracker we used in our experiment generates the pixel coordinates of the gaze point of the participants. ArUco markers [9] can be placed in the environment in order to help map gaze coordinates to physical locations (See Figure 1). These markers are robust and perform well when they are viewed from an angle or in the case of occlusion. We implemented ArUco marker detection in Python using OpenCV. Each ArUco marker has a unique pattern which corresponds to an ID from 0 to 1024. We used an online tool to generate the markers.

3.2 Determining AOIs for Coding Area

We placed markers in a grid pattern on the whiteboard. The size of each marker on the board is 1x1 cm and the distance between each is 9 cm. There are 15 columns and 11 rows of markers. We placed markers on the board in row-wise ascending order. We selected the grid design for several reasons. First, to account for subjects changing their distance from the whiteboard (getting too close), we ensured the distance between markers was close enough to always be visible in the video recordings. Second, the grid pattern enabled us to easily calculate gaze regressions in our analysis.

For the paper setting, we used 0.5x0.5 cm ArUco markers printed in a Legal page layout. We placed the markers with 4.4 cm distance for top and bottom of the frame and 2 cm distance between the markers for the sides of the frame.

3.3 Calculating Gaze Regressions

For the whiteboard setting, in each video frame we determined the nearest markers to the gaze point. We considered regression as any backward movement of the gaze more than one row. For the paper setting, we mapped the pixel distances into actual centimeters since the distance of the user to the paper makes the pixel distances vary.
in each frame. After calculating the actual distances, we considered 2 cm gaze point backtrack as the regression threshold.

3.4 Pilot Methodology

We recruited 8 graduate and 3 undergraduate participants to participate in our study. First, we helped participants don and calibrate the SMI head-mounted eye-tracking glasses. The glasses were connected to a mobile phone placed in the participant’s pocket. Next, we assigned participants two tasks of comparable difficulty from “Elements of Programming Interviews in Java” [2]: (1) reversing all words in an input string, and (2) testing a string for being a palindrome. We allowed participants to write code to complete each task in two settings: on a whiteboard and on paper. The order of settings varied randomly across participants to account for learning and ordering effects. Each participant completed both tasks in less than 45 minutes. We then conducted debriefing interviews with each participant. We collected and analyzed 12 measurements of eye tracking data from each participant to analyze stress and cognitive load.

4 PRELIMINARY RESULTS

To identify statistically significant differences between the whiteboard and paper setting, we performed a Wilcoxon signed-rank test. Our results indicated that 6 eye-tracking measurements reveal the differences between the two settings (see Table 1). Statistically significant p-values has been shown in green, with median values highlighted in yellow. We found no statistical differences in any measure or time performance between the two tasks.

### Table 1: Statistical significant differences (p-value<0.05) of various summary eye measures between different interview settings.

<table>
<thead>
<tr>
<th>Eye measure</th>
<th>Whiteboard</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>median</td>
</tr>
<tr>
<td>Trial duration [s]</td>
<td>405.3</td>
<td>407.5</td>
</tr>
<tr>
<td>Visual Intake Frequency [count/s]</td>
<td>1.94</td>
<td>2.2</td>
</tr>
<tr>
<td>Visual Intake Duration Average [ms]</td>
<td>276.8</td>
<td>295.4</td>
</tr>
<tr>
<td>Visual Intake Dispersion Average [px]</td>
<td>132.2</td>
<td>117.7</td>
</tr>
<tr>
<td>Saccade Frequency [count/s]</td>
<td>1.73</td>
<td>1.9</td>
</tr>
<tr>
<td>Saccade Duration Average [ms]</td>
<td>375.8</td>
<td>96.9</td>
</tr>
<tr>
<td>Saccade Amplitude Average [Â°]</td>
<td>5.84</td>
<td>5.3</td>
</tr>
<tr>
<td>Saccade Velocity Average [Â° /s]</td>
<td>2326.5</td>
<td>213.3</td>
</tr>
<tr>
<td>Saccade Latency Average [ms]</td>
<td>1432.3</td>
<td>356.1</td>
</tr>
<tr>
<td>Blink Frequency [count/s]</td>
<td>0.12</td>
<td>0.1</td>
</tr>
<tr>
<td>Blink Duration Average [ms]</td>
<td>1330.4</td>
<td>571.8</td>
</tr>
<tr>
<td>Regression Frequency [count/s]</td>
<td>0.15</td>
<td>0.14</td>
</tr>
</tbody>
</table>

1 Whiteboard predominately has statistical differences in measures associated with higher cognitive load.

2 High cognitive load associated with higher regressions, higher fixation dispersion, longer blinks, larger saccade velocity, longer saccade duration. Stress associated with shorter fixation duration, larger saccade velocity, and longer saccade duration.

Under pressure. We observed significantly shorter duration for fixations when participants were solving problems on the whiteboard. We believe that the whiteboard setting may have placed participants under pressure to keep shorter periods of attention [10, 23]. As a result, this may limit the ability for a participant to reflect and reason during problem solving. In contrast, participants finished their tasks on the paper in 50 seconds on average. After the experiment, most participants rated the whiteboard setting as more stressful. We also observed participants display several visible nervous tics, such as humming and face and hand twitches, that were only noticeable when solving problems on the whiteboard. Finally, it is interesting to consider that for some problem-solving tasks, limited amounts of stress can enhance performance to a limited degree [25]. In some cases, whiteboard interviews can enhance performance for some candidates; however, by increasing difficulty or time pressure, these effects can be quickly reversed.

Higher cognitive load. We observed several measures related to higher cognitive load. When solving problems on the whiteboard, participants had 3x as many regressions versus paper. During cognitively demanding or stressful tasks, a programmer may have more difficulty sustaining attention in working memory. Regression frequency shows that how many times the participant needed to look back to their previous lines of code, which can indicate memory failure as well as higher uncertainty. Higher saccade duration average and saccade velocity average in the whiteboard setting indicate higher stress and lower concentration while doing the task. Finally, higher blink duration average happens during complex cognitive process. During whiteboard problem-solving, we observe a 2x increase in blink duration, which indicates that the participants experienced higher cognitive load in the whiteboard.
setting. Some of these measures are consistent with stress caused by the public performance of the task; however, other measures may have resulted from the larger coding area of the whiteboard. For example, higher saccade velocity and duration may be consistent with increased difficulty in locating code fragments over a large space, which is less likely to occur in a paper setting.

5 VISION: INCLUSIVE INTERVIEWS

Our preliminary results demonstrate that our approach is a promising way to understand the impact of technical interviews. However, more studies are needed to better understand what characteristics contribute to high cognitive load. We highlight some future steps:

- Simple vs. Complex tasks on a whiteboard
- Solving problems on whiteboard, but in a private room
- Effect of interviewer interruption
- Effect of previous problem practice

The first study would allow us to isolate factors related to problem solving problem on a large space. The second study would allow us to study the effect of public performance. The third study would allow us to study how interruptions may further add stress and high cognitive load. The fourth study examines the effect of previous practice on reducing performance anxiety. In addition to future studies, collecting alternative measures of information, such as heart rate variability, can offer deeper insight into the cognitive state of a programmer. Finally, we can analyze in more detail how problem strategies and solutions change with interview settings.

Once we better understand the impact of certain interview practices, we can then design alternative procedures or interventions that reduce unnecessary cognitive load in candidates. We will explore a set of interventions and evaluate their impact on cognitive load. One example intervention includes the concept of an interview “blackout” [7]. For example, an interviewer might say, “now that I have explained the problem, I will step out for about 4 minutes to allow you to digest the problem.” This simple measure can allow candidates to reflect on a problem in isolation, potentially reducing anxiety and allowing the candidate the opportunity to reflect on potential approaches uninhibited.

Finally, we can derive a recommended set of interview practices that have been validated in terms of enabling assessment of the candidate while minimizing unnecessary cognitive load.

6 CONCLUSION

Technical interviews provide visibility into a candidate’s cognitive state and thought processes. Programming is a cognitive intensive task that defies expectations of constant feedback that today’s interview processes follow. This has left a gap in understanding what goes on during the programming interview process and how to properly assess programming skills of candidates to succeed at these interviews. With the proposed approach, we will begin to comprehend the cognitive state and sustained attention of candidates during technical interviews to refine the interview process.

ACKNOWLEDGEMENT

This material is based in part upon work supported by the National Science Foundation under grant number 1559593.

REFERENCES