An Eye-Tracking Visualization Towards Predicting User Distraction

CPSC 547 Project Proposal

Shareen Mahmud
shareenm@cs.ubc.ca

Jan Pilzer
pilzer@cs.ubc.ca

Vanessa Putnam
vputnam@cs.ubc.ca

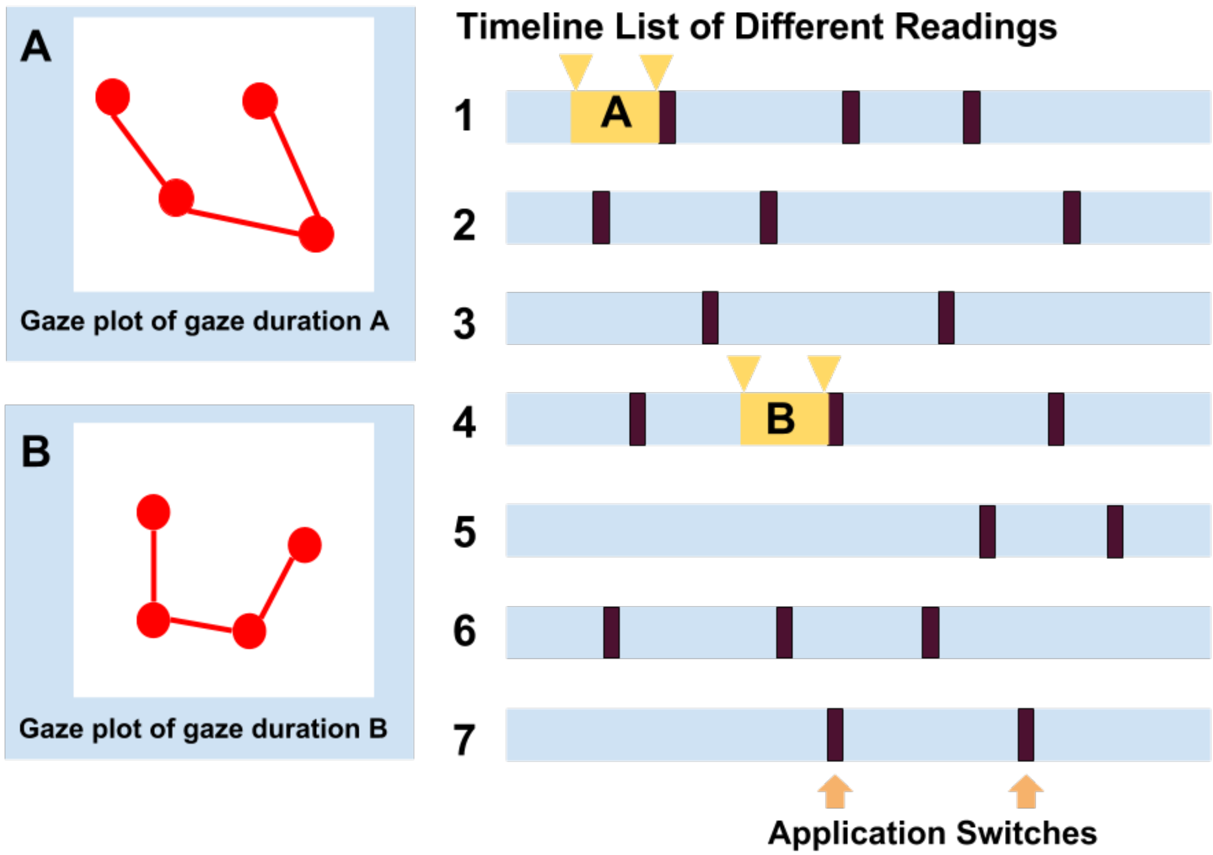

Fig. 1. Overview and two selected events.

1 Introduction

We propose an eye tracking data visualization that will allow the users to analyze durations of time before a reader distraction occurs. Our data collection will be gathered through an existing application known as Gaze Reader. This application is a tool for reading on a computer and has the ability to track a reader's gaze pattern, with the help of a Tobii eye tracker, and record a reader's change from the gaze reader to another application. We define this change as an application switch. Our goal is to create a visualization of this data in order to understand the gaze pattern before a reader distraction occurs. We define a distraction to be either a switch from Gaze Reader to an application not reading related, or when a reader has been inactive with no eye tracking data collected. We are specifically interested in the threshold of time before distraction occurs. From our methods we will be able to deduce the length of time before this distraction as well as the patterns of gaze associated with distraction.

2 Personal Expertise

The motivation and data for this project originate from a project in another course, Sensing and Supporting Developer Productivity. We built an application to capture eye tracking data of participants while they read their research papers. Given the mandatory nature of these papers and the claimed short attention span of current students, we expect to see distractions caused by self-interruptions. As these self-interruptions are even more costly than external interruptions [3], our goal is to find patterns which can be used to predict upcoming self-interruptions. In addition to the motivation for this project, our group also has prior expertise in human computer interaction, machine learning, as well as strong programming backgrounds.

3 Data and Tasks

Domain: Eye Tracking is a technique that measures an individuals eye movement in order to understand where one is looking. Eye movement data can be used to identify fixation points which tell us where a persons gaze lingers for a certain time. In addition, the data consists of saccades which indicate the points at which a person rapidly looks from one position to another. Therefore, eye tracking can be used to gain insights on peoples behavior as they interact with a system.

Table 1. Data Types

Data Type	Description
Item	Application switch
Attributes	User: categorical
	Reading: categorical
	Gaze, Fixation: The range of gaze and fixation locations depend on the screen resolution in pixels.
	Head pose: The range of head position is between -1000 to 1000 mm.
Derived Attributes	Saccade
	Saccade speed

The goal of our visualization is to enable the users to look at multiple application switches and to understand a persons gaze pattern before an application switch occurs. This gaze pattern can then be used to infer distraction.

Our tool will accomplish this task by allowing the users to carry out the following actions:

Analyze: Users will be presented with eye tracking data from the Gaze Reader application which they can use to discover patterns within and between application switches

Search: Users can search for gaze patterns of interest in order to deduce distraction. They can browse through a series of application switches on a timeline, choose a location and look at the gaze patterns corresponding to that location.

Query: Users can look at multiple gaze durations of the same length on the timeline and compare this between application switches for each reading or for different readings and readers. They can also compare between two different visualizations (such as a gaze plot or a heatmap) of the same duration.

The target for every action discussed above is an application switch. In order to help the users infer distraction, there will be a predicted gaze plot showing distraction pattern for every gaze duration.

Scale: We are currently aiming to collect eye tracking data of at least 15 participants with a minimum of one reading each and with multiple application switches. The exact number of application switches will depend on the participants attention span. The dataset is static.

Our dataset is a table that contains items and attributes. We cover the details of these items and attributes in Table 1.

4 Proposed Solution

Our goal for this visualization is to understand the pattern of gaze associated with distraction using the gaze and application switch data collected from Gaze Reader. We will make use of several idioms for this visual encoding including: a timeline, multiple views, heatmap, gaze plot, 3D head pose, and prediction.

To appropriately encode the full gaze duration and associated application switches we will utilize a timeline of the data gaze duration from start to finish. Within each timeline bar we will use notches to indicate when an application switch has occurred. Initially, we will try a timeline that uses the raw gaze duration data, however, as our project continues we may employ a scaled or normalized approach to leave out data that may not provide distraction information. Similarly, we may also try an encoding that leaves out the timeline and only provides application switches and data for a fixed duration of time.

From the timeline, users will be able to select a duration of time, via a sliding window, to view in detail. We call this duration selection a gaze duration, and the associated view a gaze view. We would like to implement a visualization that will allow users to select multiple gaze durations and display multiple associated gaze views. In order to allow for a detailed analysis of the data provided, we will utilize a few different idioms that viewers can select from. We would like to include a heatmap of fixations, a gaze plot, and a 3D rendering of head pose for each gaze duration selected. This multiple view approach will allow users to compare different gaze durations by analyzing the different idioms of multiple gaze views. A visualization of this idea is shown in Figure 1.

In addition to the idioms already mentioned, we also propose a prediction step. We will gather a training set that includes manually tagged distraction points within the set of application switches. This set will be learned with off the shelf machine learning tools and applied to a test set with untagged application switches. Our goal is to come up with a model that represents the pattern associated with a gaze view for distraction. This will allow a user to compare a gaze view against a prediction of what we believe distraction to look like. Our hope is to allow the users to deduce if the gaze view is a distraction or not based off this comparison.

5 Scenario

The visualization solution proposed here is going to be used by the Gaze Reader team to explore and make sense of the large amounts of data recorded during the eye tracking experiments. We will therefore describe the scenario of use from the perspective of Sam, who is Jans teammate in that project.

To start Sam looks at the overview to see the list of available readings. If he is looking for a specific instance, he can easily find what he is interested in because the application switches can be sorted by readings or grouped by participants. As he is trying to find patterns to predict distractions, only the lead-up to a distraction event is of interest. The list in the overview has these events clearly marked so that Sam can easily find and select one. He brushes to select the exact duration he wishes to see in detail. To compare, he selects further application switches.

On the comparison pane, he sees multiple gaze views, each corresponding to one of the selected distraction events. The views visualize one property of the data streams. Initially, he is seeing the gaze plot over time. As they prove inconclusive, he decides to switch to a view of heat map. Additionally, he adds visualizations for the reading speed to each of the views.

6 Implementation Approach

The visualization will be implemented as a web application. The popularity of web technologies for user interfaces has increased significantly in recent years [4] and many libraries are available for providing visualization in web applications. For the core visualizations we plan to use d3.js[1] with several d3.js plugins[2] which contain all required components for loading, parsing, and formatting data, and rendering the visualization.

Given our desire for an interactive system with linked views, a more general library for user interfaces might be used. Furthermore, a design framework such as Bootstrap[3] could be used to decrease overall styling effort.

7 Milestones and Schedule

We are prepared to spend about 190 hours together towards this project. Table 2 provides a breakdown of the project's tasks. The pitches were delivered by each team member independently. The proposal, the preparation of the project reviews, and the materials for the final submission will be worked on together. The implementation tasks outlined in the schedule build on each other so a distribution between team members is not feasible. We will split the tasks into smaller, actionable issues and distribute those according to availability as we progress with the overall implementation.

[1] https://d3js.org/
[2] https://github.com/d3/d3/wiki/Plugins
[3] https://getbootstrap.com

Table 2. Task Schedule.

Task	Est Hours	Deadline	Description
Pitch (x3)	12	Oct. 17	Create slides, rehearse pitches.
Proposal	24	Nov. 5	Discuss project ideas, create mockups, write proposal.
Project Review 1	2	Nov. 20	Prepare slides.
Project Review 2	2	Dec. 5	Prepare slides, have some version of demo ready.
Implementation		Dec. 12	Completed vis tool.
- Prepare Data	8	Nov. 10	Tag, clean, and filter existing data.
- Application structure	16	Nov. 17	Set up project frontend. Create vis controls.
- Overview Vis	32	Nov. 27	Implement overview list of readings and gaze durations.
- Selection functionality	32	Dec. 3	Implement gaze duration selection and detail comparison functionality.
- Detail Vis	32	Dec. 10	Implement detail visualizations.
Presentation	10	Dec. 12	Prepare slides, demo, and potential video. Rehearse presentation.
Final paper	20	Dec. 15	Finalize paper. Draft to be written Dec. 3-13.

8 Previous Work

The importance of visualization for a successful analysis of eye tracking data has been confirmed in different contexts [5,6]. These visualizations typically focus on areas of interest (AOI) that participants look at [2]. The project whose data we are visualizing however tries to find universal patterns that apply independent of the current display space. We plan to include standard gaze plots and scanpaths [1] to determine their effectiveness for this use case. Additionally, we include visualizations for derived metrics like reading speed.

References

[1] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf. Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2889–2898, Dec. 2012.

[2] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. State-of-the-Art of Visualization for Eye Tracking Data. In R. Borgo, R. Maciejewski, and I. Viola, editors, *EuroVis - STARs*. The Eurographics Association, 2014.

[3] I. Katidioti, J. P. Borst, M. K. van Vugt, and N. A. Taatgen. Interrupt me. *Comput. Hum. Behav.*, 63(C):906–915, Oct. 2016.

[4] M. J. Rees. Evolving the browser towards a standard user interface architecture. *Aust. Comput. Sci. Commun.*, 24(4):1–7, Jan. 2002.

[5] B. Sharif, M. Falcone, and J. I. Maletic. An eye-tracking study on the role of scan time in finding source code defects. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 381–384, New York, NY, USA, 2012. ACM.

[6] H. Uwano, M. Nakamura, A. Monden, and K.-i. Matsumoto. Analyzing individual performance of source code review using reviewers' eye movement. In *Proceedings of the 2006 Symposium on Eye Tracking Research &Amp; Applications*, ETRA '06, pages 133–140, New York, NY, USA, 2006. ACM.