ETGraph: A Graph-Based Approach for Visual Analytics of Eye-Tracking Data

Yi Gu[a], Chaoli Wang[b], Robert Bixler[a], Sidney D'Mello[a,b]

[a] *Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556*
[b] *Department of Psychology, University of Notre Dame, Notre Dame, IN 46556*

Abstract

Mind wander(ing) (MW) or zoning out is a ubiquitous phenomenon where attention involuntary shifts from task-related processing to task-unrelated thoughts. Unfortunately, MW is a highly internal state so it cannot be readily inferred from overt behaviors and expressions. To help experts investigate mind wanderings, we present a graph-based approach for visual analytics of eye-tracking data, which utilizes the graph representations to illustrate the reading patterns and further help experts detect and verify mind wanderings based on the graph structures and other graph attributes. The input data are collected from multiple participants reading multiple pages of a book on a computer screen. Our approach first clusters fixations into fixation clusters, then creates the eye-tracking graph, i.e., ETGraph, for use in conjunction with the standard page view, time view, and statistics view. The graph view presents a visual representation of the actual reading patterns of a single participant or multiple participants and therefore serves as the main visual interface for exploration and navigation. We design a suite of techniques to help users identify common reading patterns and outliers for analytical reasoning at three different levels of detail: single participant single page, single participant multiple pages, and multiple participants single page. Interactive querying and filtering functions are provided for reducing visual clutter in the visualization and enabling users to answer questions and glean insights. Our tool also facilitates the detection and verification of mind wandering that the experts seek to investigate. We conduct a user study and an expert evaluation to assess the effectiveness of ETGraph in terms of its visual summarization and comparison capabilities.

Keywords: Eye-tracking data, Visual analytics, Graph layout, Saccade outlier detection, Repeated scanpath detection, Participant comparison and clustering

1. Introduction

With advances of the eye-tracking technology, eye-trackers are getting increasingly affordable for use in research and education. In this work, we study eye-tracking data collected from multiple participants reading multiple pages of a book on a computer screen. A research group led by a cognitive scientist collected the data in order to investigate cognitive processes during reading. In this paper, we focus on attentional lapses called mind wandering, but our solution can be applied to investigate other cognitive and affective phenomena, such as cognitive load, inference generation, boredom, and so on.

Mind wander(ing) (MW) or zoning out is a ubiquitous phenomenon where attention involuntary shifts from task-related processing to task-unrelated thoughts [1]. Considerable research over the last 5-10 years has documented the widespread incidence and negative consequence of MW both in the lab and in the real world. In one highly-cited, large-scale study, MW was tracked in 5,000 individuals from 83 countries working in 86 occupations with an iPhone app that prompted people to report MW at random intervals throughout the day [2]. People reported MW for 46.9% of the prompts, which confirmed numerous lab studies on the pervasiveness of MW [3, 4]. MW is also more than merely incidental as a recent meta-analysis of 88 studies indicated a negative correlation between MW and performance across a variety of tasks [5], a correlation which increases in proportion to task complexity. MW occurs around 30% of the time during reading and is negatively correlated with reading comprehension.

Unfortunately, MW is a highly internal state so it cannot be readily inferred from overt behaviors and expressions [6]. Thus, the most common way to measure MW is via self-report. Self-caught methods ask people to monitor their attentional levels and to indicate (e.g., by pressing a key) when they catch themselves MW. For example, a participant in a reading study may be asked to press a key when they realize that "they have no idea what they just read because they were thinking about something else altogether" [7]. The same instructions are used in probe-caught methods; however, participants are prompted (e.g., via an auditory probe) at multiple intervals to indicate if they are MW at the time of the probe [8]. MW data collected in this fashion have shown predictable relationships with physiology [9], pupillometry [10], eye gaze [7], and task performance [5], thereby providing some validity for this measurement approach. However, there are many limitations of self-report measures, so it would be beneficial to obtain behavioral indicators of MW. In this paper, we focus on eye gaze to track MW, which is motivated by decades of scientific evidence in support of an eye-mind link that suggests a tight coupling between internal thoughts and eye movements [11]. Our goal in this work is to design a visual interface that helps researchers investigate reading patterns (adduced from eye-movements) associated with MW. Our long-term goal is to use these expert insights to improve automated measures of MW, which are still in their infancy [12, 13].

We restrict our attention to the reading study with static stimuli (i.e., static text on screen) and aim to investigate reading behaviors for tens of participants. In a recent article, Raschke et al. [14] pointed out that visually analyzing multiple viewers with an individual stimulus is an interesting research topic. It is also challenging to present an effective solution to find patterns, detect outliers, and compare different participants. The key issue is how to design a visual analytics tool that leverages different visual mappings, interfaces and interactions to facilitate visual exploration, navigation and comparison of the vast amount of eye-tracking data.

Our main contribution lies in the designing of a visual analytics framework that helps researchers investigate reading patterns, which could be further categorized into three different levels of detail: SPSP (single participant single page), SPMP (single participant multiple pages), and MPSP (multiple participants single page). For SPSP, our visual interface allows researchers to capture the normal and abnormal reading patterns of a participant on a single page and identify possible MWs. This may be used to improve the automated measures of MWs. For SPMP, our visual interface helps researchers identify similar behaviors among continuous pages. For MPSP, the common reading patterns of the same page from all participants are illustrated. In addition, we allow users to compare the differences between any two selected participants.

To this end, we propose to transform the eye-tracking data gathered from a reading study into a graph view for visual analytics. Graph-based representations have been utilized for eye-tracking data analysis. For instance, Tory et al. [15] studied the relation between areas of interest (AOIs) using a directed graph visualization. In such a graph, each node represents one AOI and an edge connecting two nodes represents their transition. The edge thickness depicts the number of transitions between the two AOIs. In their work, the graph view was used mainly for a visual overview but not for interactive exploration. In contrast, our work is pitched at a finer level of detail. That is, instead of using AOIs for visual summarization, we group fixations into clusters and build a graph, i.e., ETGraph (eye-tracking graph), to support interactive examination of the underlying structure in the eye-tracking data. Multiple coordinated views are utilized to dynamically link the graph view with the standard page view during the interaction.

We design a suite of techniques to help users identify common reading patterns and outliers for analytical reasoning at different levels of detail. Our tool enables visual comparison of different pages being read by a single participant as well as when the same page is read by different participants. It also supports a global overview of reading patterns of all pages by all participants and local exploration of a single page being read by a single participant. We demonstrate the effectiveness of our approach by showing experimental results gathered from analyzing the eye-tracking data. We also report the feedback of using our tool for visual exploration and MW investigation.

2. Related Work

Rayner [11] synthesized over 100 years of eye-tracking research and conducted an excellent survey of eye-tracking applications in reading and other information processing tasks. Duchowski [16] presented a breadth-first survey of eye-tracking applications in the following domains: neuroscience, psychology, industrial engineering and human factors, marketing or advertising, and computer science. Recently, Blascheck et al. [17] presented a comprehensive state-of-the-art report on techniques for visualizing eye-tracking data. They classified the visualization techniques into different categories based on properties of eye-tracking data and properties of visualization techniques.

Tracking eye-movement leads to vast amounts of fixation points and scanpaths which can be clustered and visualized for clear observation of patterns or outliers. Santella and DeCarlo [18] presented a robust clustering of eye-movement recordings using the mean-shift method, which forms a structured representation of the viewer's attention and avoids heavy influence from noise or outliers. Špakov and Räihä [19] introduced EiKV, which shows the reading and typing processes in parallel with details for each word presented in word bars so that users could identify the unusual events. Goldberg and Helfman [20] proposed a solution to identify scanning strategies by automatically aggregating groups of matching scanpaths. First, they converted each scanpath into a sequence of AOIs visited in order. Sequences of AOIs were concatenated into one sequence and plotted with a dotplot. Then they used linear repeated scanpaths to find matching sequences in the dotplot for clustering the scanpaths hierarchically. Tang et al. [21] designed EyeMap, a system which supports word segmentation, eye movement data visualization, and XML data format. Since word segmentation could identify separated words so that fixations are mapped to the words, EyeMap could support writing systems using different languages. Furthermore, gaze, scanpath, and statistics information are displayed to support various kinds of queries. In addition, the XML data format is utilized for describing data from a wide range of reading experiments for data export and sharing.

To visualize the spatiotemporal behaviors of eye-movement data, one can use heat maps or gaze plots. However, these visual representations suffer from high aggregation (heat maps) and overplotting (gaze plots). New visual mappings and representations are needed for investigating the vast amounts of spatiotemporal eye gaze trajectories. Tsang et al. [22] presented eSeeTrack, an eye-tracking visualization prototype to facilitate the exploration and comparison of sequential gaze orderings in a static or dynamic scene. Their work integrates a timeline and a tree-structured representation to encode multiple aspects (duration, frequency, and fixation ordering) of eye-tracking data. Burch et al. [23] transformed eye-movement data into a dynamic graph and achieved a fair tradeoff between aggregation and details. Their dynamic graph is a sequence of static graphs where nodes represent AOIs and directed edges show transitions between source and target AOIs. Burch et al. [24] designed AOI Rivers for investigating time-varying fixation frequencies, transitions between AOIs, and the sequential order of gaze visits to AOIs. Based on the ThemeRiver technique, they represented the trajectory data as time-varying river-like structures enhanced by influents, effluents, and AOIs transitions, similar to Sankey diagrams.

Beyond analyzing eye-tracking data, eye-movement analysis has gained its popularity as a tool for evaluating visualization research. Andrienko et al. [25] proposed a visual analytics methodology originated from analysis of geographic data for analyzing large amounts of eye-tracking data. They focused on deriving common task solution strategies for a given static stimulus shown to participants. Their work presents a systematic evaluation of movement analysis methods for the applicability of eye-tracking data and provides the guidelines for choosing appropriate methods given the analysis goals. Blascheck et al. [26] presented a visual analytics approach for an integrated analysis of multiple concurrent evaluation procedures such as measures of task performance, think-aloud protocols, analysis of interaction logs, and eye tracking. An efficient exploratory search and reasoning process is supported through automatic pattern finding to derive common eye-interaction-thinking patterns between participants.

3. Research Questions

Blascheck et al. [17] defined the basic terminology related to eye-tracking data. We briefly introduce several of them that are used in this work. First of all, *gaze points* are the raw eye-tracking data and each *fixation* is an aggregation of gaze points based on specified area and timespan. Furthermore, *saccades* describe a rapid eye movement from one fixation to another, and a *scanpath* is a sequence of alternating fixations and saccades. Analyzing them would help users understand the eye-movements, therefore, there are a lot of related research questions. Since our eye-tracking data are gathered from multiple participants reading multiple pages of a book, we propose the following research questions categorized into three different levels of detail: SPSP (single participant single page), SPMP (single participant multiple pages), and MPSP (multiple participants single page). The questions associated with SPSP, SPMP, and MPSP focus on the reading patterns of a participant reading one page, the consistency of a participant reading multiple pages, and the behavior similarities/differences between participants, respectively.

- SPSP (single participant single page):
  - **Q1.** What is the scanpath structure of each participant when reading a single page?
  - **Q2.** Does the participant exert a different amount of effort reading different parts of the page?
  - **Q3.** Does the scanpath involve forward and/or backward saccade outliers (i.e., saccades with amplitudes larger than a given threshold)? If yes, when and where do these saccade outliers occur and how frequent are they? Does the same saccade outlier occur multiple times?
  - **Q4.** Does the scanpath involve repeated scanpaths (i.e., a scanpath that represents rereading previously read text along the same path)? If yes, when and where do these repeated scanpaths occur and how frequent are they? Does the appearance of saccade outliers have any correlation with the appearance of repeated scanpaths?

	C1	C2	C3	C4	C5	C6
SPSP	**Q1**	**Q3**	**Q4**	**Q5**	-	**Q2**
SPMP	**Q6**	-	-	-	-	**Q7**
MPSP	**Q8**	-	-	-	**Q10**	**Q9**

Table 1: The ten research questions **Q1** ∼ **Q10** associated with SPSP (single participant single page), SPMP (single participant multiple pages), and MPSP (multiple participants single page) are classified into six categories **C1** ∼ **C6**.

  - **Q5.** Does the participant MW on a page? If yes, when and where does MW occur?
- SPMP (single participant multiple pages):
  - **Q6.** Is the reading pattern of a participant consistent across all pages?
  - **Q7.** What are the temporal dynamics of reading behavior across consecutive pages? For example, do the saccade outliers on the current page have a relationship with the saccade outliers on the next page?
- MPSP (multiple participants single page):
  - **Q8.** What are the common patterns of multiple participants when reading the same page?
  - **Q9.** Do they spend a different amount of time reading different parts of the page?
  - **Q10.** What are the outliers (who, when and where)? Can we cluster participants based on different reading patterns exhibited on the same page?

The research questions can be classified into six categories, as shown in Table 1: **C1** scanpath structures (**Q1**, **Q6**, **Q8**), **C2** saccade outliers (**Q3**), **C3** repeated scanpaths (**Q4**), **C4** MW (**Q5**), **C5** participant clustering (**Q10**), and **C6** reading efforts (**Q2**, **Q7**, **Q9**). To better answer these questions, we introduce four views as shown in Figure 1: page view, graph view, time view, and statistics view. Categories 1 to 4 can be answered using the graph, page, and time views. Categories 5 and 6 can be answered using the statistics view. In addition, these four views can be combined together to help users better explore and understand the data.

4. ETGraph Construction

Our goal is to design a visual analytics framework that can help users understand reading patterns of the participants, identify the anomalous behaviors, and group participants. Specifically, how can we apply the three levels of analysis to obtain new insights on reading patterns? Can we visually discriminate reading patterns from the graph representations? Can we find

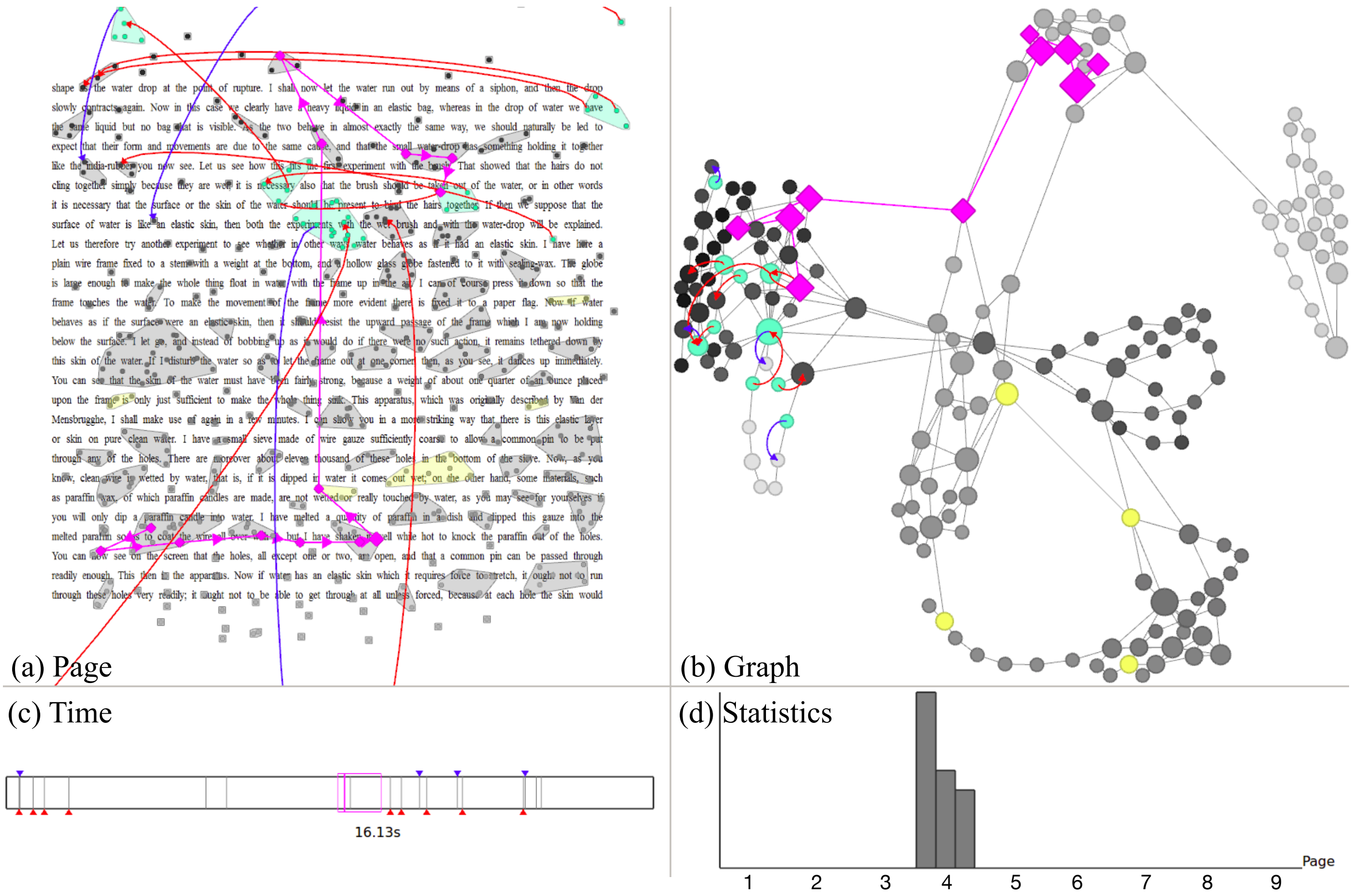


Figure 1: The four views of ETGraph. (a) page view, (b) graph view, (c) time view, and (d) statistics view show saccade outliers and MWs of SPSP. In (a) and (b), fixation clusters and nodes corresponding to saccade outliers are highlighted in yellow. The selected clusters and nodes are in green. Red and blue edges indicate backward and forward saccade outliers, respectively. The fixations and nodes around MW are highlighted as pink diamonds. The arrows between the diamond fixations in the page view shows the scanpaths around MW. In (b), the shading of nodes (from dark to light) indicates the presumed reading order. In (c), the vertical lines indicates the time points that the saccade outliers occurred. Red and blue arrows correspond to the selected saccade outliers. The pink rectangle and the number below show the MW and its duration. (d) shows the distribution of the numbers of saccade outliers in page sections (each page is partitioned into three equal sections: top, middle and bottom).

out the relations between saccade outliers and repeated scanpaths? Furthermore, can we visually figure out the similarities and/or differences between the reading patterns on different pages or between different participants? Finally, can we embed the visualization of MWs to help scientists analyze them?

Our strategy to analyze the given eye-tracking data is to first cluster fixations into fixation clusters to produce a coarse-level representation of the data. There are two benefits to doing this. First, our fixation clustering provides spatial closeness. Fixation clusters are spatially close fixations which are similar to gazes as defined in Blascheck et al. [17]. However, unlike gazes, we do not consider the temporal ordering of fixations when we perform clustering. Second, fixation clusters are not as coarse as AOIs which represent regions of specific interest on the stimulus. Fixation clusters can be created for a single page using fixations from a single participant or by combining all the fixations from multiple participants. Combining all fixations for clustering would allow us to visualize the reading patterns of different participants with a common ground of the same fixation clusters.

We propose a graph-based representation for analyzing eye movements using fixation clusters as nodes and a set of saccades as a directed edge between nodes. We call this visual representation the *ETGraph*, i.e., eye-tracking graph. Visualizing such a graph can be achieved using force-directed graph layout algorithms or projection-based methods such as multidimensional scaling. In the original page view, fixations and saccades can be plotted to produce scanpaths. However, nodes are solely constrained by their spatial locations on the page. With this stringent constraint, large saccadic amplitudes may not always be of interest (e.g., a saccade moves from the end of one line to the beginning of the next line). Unlike the page view, nodes in the graph view are not constrained by their corresponding spatial locations of fixation clusters and the graph structure are dictated by node connectivity (i.e., the actual reading). Therefore, the graph view can reveal the underlying nature of the reading pattern.

Finally, we design ETGraph so that users can smoothly transit between SPSP, SPMP, and MPSP. This facilitates the examination of reading patterns from both global and local perspec-

Algorithm 1 CLUSTER INDICES Idx = FINDCLUSTERS(V)

Create a temporary point set P to store the locations of input points V after each movement
for each point p_i in P **do**
  $p_i = v_i$
for each iteration j **do**
  for each point p_i in P **do**
    $p_i^j = S(p_i^{j-1})$
Create an empty point set C to store the centroids of clusters
for each point p_i in P **do**
  if the location of p_i is not identical to any point in C **then**
    Add p_i to C
Create a set Idx to store the cluster indices
for each point p_i in P **do**
  for each point c_j in C **do**
    if the location of p_i is identical to c_j **then**
      $idx_i = j$
return Idx

tives while making connections between them.

4.1. Mean-Shift Algorithm

We use the mean-shift algorithm, a density-based clustering to group fixations. This algorithm is deterministic and robust to outliers. In addition, it does not require users to input the number of clusters. Santella and DeCarlo [18] also demonstrated the mean-shift method produces better quality of clusters compared to the k-means clustering or expectation maximization (EM) algorithms. In the mean-shift algorithm, the input points are moved to a denser configuration so that they are naturally grouped into clusters. Moving each point p to a new location is based on the locations of its neighbors

$$S(p) = \frac{\sum_j \ker(p - p_j) p_j}{\sum_j \ker(p - p_j)}, \tag{1}$$

where the kernel function estimates the contribution of each neighbor p_j, i.e.,

$$\ker(d) = \exp\left(\frac{-d_x{}^2 - d_y{}^2}{\epsilon^2}\right), \tag{2}$$

where d is the Euclidean distance between p and p_j, and d_x and d_y are the projections of d along the x and y directions, respectively. ϵ is a given threshold and the points with a distance to p larger than ϵ are not considered. Since our data are collected from text-reading experiments, the x and y directions are related to word length and line spacing, respectively. We therefore revise the kernel function to

$$\ker(d) = \exp\left(\frac{-d_x{}^2}{\epsilon_x^2}\right) \times \exp\left(\frac{-d_y{}^2}{\epsilon_y^2}\right), \tag{3}$$

where ϵ_x is the average word length and ϵ_y is the line spacing.

Algorithm 1 first moves the input points V to a denser configuration P. This process stops when the number of iterations reaches a user-defined threshold or the locations of points in P do not change. The points in P that move to the same location become a cluster. The locations are treated as the centroids of clusters. We assign the points to clusters by measuring the distances between the points and the centroids.

4.2. Transition Graph

After clustering all fixations of each page using the mean-shift algorithm, we construct a transition graph. Each node in the graph denotes a fixation cluster. A directed edge between two nodes represents a transition. A transition $i \rightarrow j$ occurs between two clusters i and j if there is a saccade from a fixation in i to another fixation in j. The transition frequency $f_{i\rightarrow j}$ is the number of transitions from i to j. The directional transition probability $p_{i\rightarrow j}$ is the proportion between $f_{i\rightarrow j}$ and the total number of transitions from i to all the clusters (including itself). As such, a transition indicates a chance for one cluster to transfer to another, and its probability measures how high the chance is. To draw the transition graph, we modify the Fruchterman-Reingold algorithm [27] by considering the transition probability when computing the attractive forces. Therefore, two nodes with strong transitions are placed close to each other. However, the nodes may overlap with one another due to their sizes in the drawing. To reduce the overlap while preserving the overall graph structure, we follow the layout adjustment solution given by Gu and Wang [28] which first triangulates the graph and then applies four additional forces (bidirectional, unidirectional, spring, and attractive forces).

5. SPSP, SPMP, and MPSP

ETGraph helps users identify common reading patterns and outliers for analytical reasoning at three different levels of detail: SPSP, SPMP, and MPSP. First, SPSP provides detailed examination of the reading patterns for one participant reading one page. Second, extending single page to multiple pages, SPMP visualizes the reading patterns for one participant reading continuous pages. This allows users to identify abnormal behaviors across different pages which may, for instance, indicate that the difficulty levels of some pages are different from others. Third, extending single participant to multiple participants, MPSP aims to analyze the common reading pattern and different reading behaviors among participants.

5.1. SPSP

To help users better understand detailed reading behaviors for SPSP, we provide several query functions, e.g., *saccade outlier detection*, *MW highlighting*, *graph filtering*, *path animation*, and *repeated scanpath detection*.

Saccade outlier detection automatically identifies the saccade outliers that traverse a large distance (larger than a given threshold) along the x or y direction. Users are allowed to change the threshold. By default, the thresholds along x and y directions are around 1/3 of the page width and 1/4 of the page height, respectively. These saccade outliers indicate long eye movements which may indicate abnormal reading patterns and possible MW. We further differentiate backward- and forward-reading saccades. Backward-reading saccades indicate revisiting earlier portions of the text while forward-reading saccades may indicate foreshadowing. Figure 1 shows an example of saccade outlier detection. Specifically, we show the information of saccade outliers in each of the four views.

Algorithm 2 Suffix tree T = ConstructSuffixTree(S, n)

```
Create a temporary string S′ where a unique symbol is added at the end of S
Create a tree T with an empty root
for i from 0 to n do
    substring s = S′[0, i]
    for j from 0 to i + 1 do
        if s[j, i] starts from root to a leaf edge then
            Add s[i + 1] to the leaf edge
        else if s[j, i] starts from root and ends at a non-leaf edge, but s[i + 1] is
        not the next character of the edge then
            A new leaf edge is created for s[i + 1] from the separation
return T
```

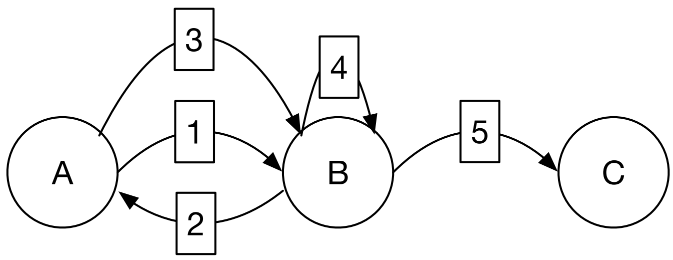


Figure 2: An example of the scanpath in the transition graph. The scanpath is $ABABBC$, the corresponding string is $ABABC$, and the repeated scanpath is AB.

MW highlighting visualizes the scanpath immediately before and after instances of MW. As shown in Figure 1 (a) and (b), the scanpath and its corresponding subgraph are highlighted in pink. This function provides a way for users to study MW in the page, graph, and time views.

Graph filtering allows users to hide nodes (fixations) and edges (saccades) that are not of interest. As users select one or a group of nodes, we automatically hide the nodes that are beyond a given distance to the selected node(s). The graph distance between two nodes is calculated using Dijkstra's algorithm. If two nodes that are not supposed to be connected in the presumed reading order are linked in the graph view, this indicates that at least one saccade outlier is present. Graph filtering also allows users to filter the graph based on time information and analyze the corresponding subgraph.

Path animation provides users the convenience of reviewing scanpath animation. We identify the starting and ending fixations f_s and f_e for the animation. By default, they are the first and last fixations on the page. However, users are allowed to select fixations from the page view or nodes from the graph view for f_s and f_e. If the user selects two nodes from the graph view, we identify the first fixation in the first node as f_s and the last fixation in the second node as f_e. If the two fixations selected do not follow the actual reading order, we swap f_s and f_e. If the user selects a node from the graph view and a fixation from the page view, we ensure that f_s occurs before f_e in the actual reading order.

Repeated scanpath detection automatically detects repeated scanpaths. We allow users to visualize them and find out their corresponding locations and time periods from the graph, page, and time views. In addition, we allow users to play back the scanpaths to compare similarities and differences in scanpaths between different pages and different participants. In order to detect repeated scanpaths, we first convert the scanpath into a string, and this string stores all the transitions between graph nodes. Since we are more interested in transitions between nodes, we ignore all the transitions within the same node. Figure 2 shows an example of the scanpath in the transition graph. We first assign IDs to the nodes. Therefore, the scanpath is $ABABBC$. Since we ignore all the self-transitions, the corresponding string becomes $ABABC$. By analyzing this string, we can detect interesting phenomenas, such as area revisitings (repeated characters, A and B), repeated scanpaths (repeated substrings, AB), and similar behaviors between two participants (common substrings of two given strings). To detect these phenomena, we utilize the suffix tree [29] to identify the repeated substrings in a given string. The suffix tree is a tree structure that stores all the suffixes of a given string. Each edge represents a substring. Therefore, all the non-leaf edges in the suffix tree represents repeated substrings. Constructing and searching takes linear time which allows for efficient queries. We first add a unique symbol \$ at the ending of the given string to become a new string. This symbol is used to indicate the ending of the given string. At each iteration, we consider a substring starting from string indices 0 to i, where i increases from 0 to $n-1$, and n is the length of the new string. Then, within each iteration, all the suffixes of the substring are inserted into the suffix tree as shown in Algorithm 2. For example, given a string ABA, A is the repeated substring. Once symbol \$ is attached to the end, the given string becomes ABA\$. According to Algorithm 2, substrings A\$, BA\$, ABA\$ would be inserted into the suffix tree. Therefore, edge A has children \$ and BA\$. Then A is identified as a repeated substring. If no such a special symbol is attached at the end of the given string, edge A would become part of ABA and thus could not be identified. To identify the common substrings of two given strings, we further add another special symbol between the two given strings to indicate the separation of the two strings and use the whole string as an input for the suffix tree. For example, given two strings BA and AA, without symbols separating them, the combined string would be $BAAA$. In this case, AA will be considered as a substring that repeats twice. However, it is not the case. If we add a special symbol *, then the combined string would be BA*AA. In this case, AA only appears once as a substring. Note that the two symbols (\$ and *) are different. Using the ending symbol for the separation of the two strings may lead to the missing of the repeated substrings. To allow users to focus on prominent substrings, we removed the substrings which are substrings of others, or less than a given length.

5.2. SPMP

To understand and compare different behaviors on different pages (SPMP), we generate a SPMP-supergraph that displays the transition graphs of all pages. An example is shown in Figure 3. The transition graphs are arranged clockwise in a spiral shape. To reduce edge crossing between pages, we first fix the positions of the first and last nodes at the middle-left and middle-right parts of each subgraph, respectively. Then we rotate each subgraph one by one to reduce the length of the edge connecting the adjacent pages.

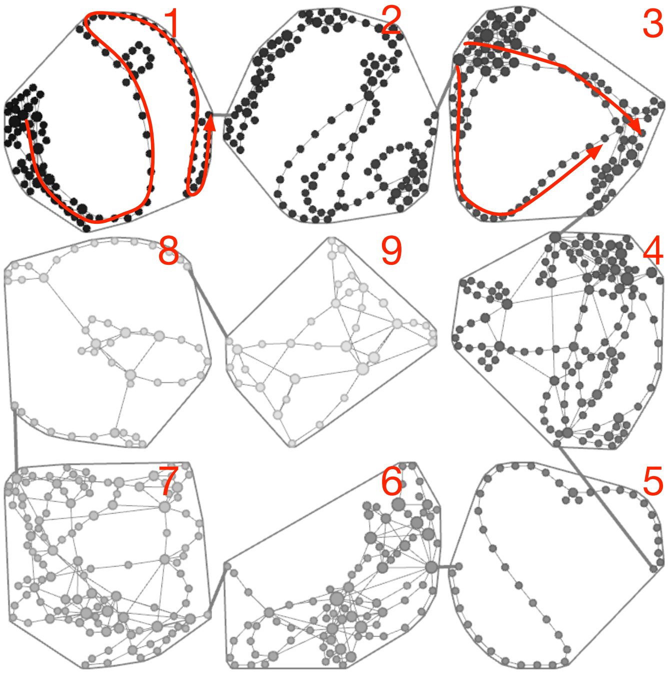


Figure 3: The SPMP-supergraph. Each subgraph corresponds to the transition graph of each page. The subgraphs are marked with their page numbers and are placed in a spiral along the clockwise order. In addition, we rotate each subgraph to reduce the length of the edge that connects two adjacent pages. The shading of nodes (from dark to light) indicates the presumed reading order.

5.3. *MPSP*

For MPSP, we cluster fixations on each page with all the fixations of all participants on that page. After clustering, we construct a MPSP-supergraph consisting of all the clusters of all participants of that page. As a result, edges with high frequencies form the major reading trend of all participants. ETGraph encodes higher frequency edges with darker gray colors so that users can easily understand the overall reading patterns. However, it is difficult to notice the trend of saccade outliers since those edges are usually drawn with lighter gray colors due to their low frequencies. To highlight saccade outliers and identify their trend, we apply the edge bundling technique [30] to bundle saccade outliers as shown in Figure 4.

It is also important to cluster and compare participants in order to analyze their similarities and differences. We provide two approaches. First, we utilize scarf plots and histograms in the time and statistics views to show explicit information (e.g., section length in milliseconds, saccade outlier distribution) for comparison. Second, we calculate the similarities between participants based on the graph information. For MPSP, the transition graph of a participant for a page is a subgraph of the MPSP-supergraph. Therefore, by comparing the similarities between two subgraphs, we can calculate the similarities between these two participants. The difference between two participants can also be calculated based on their fixation distributions, repeated scanpaths, etc. For each type of difference, we construct a distance matrix. We then normalize each distance matrix and add them together to form the final distance matrix. Finally, we utilize the k-means algorithm to cluster participants. The number

	mean shift	layout generation	layout adjustment	repeated scanpath	edge bundling
single participant (SP)	1.283	18.941	3.968	0.143	-
multiple participants (MP)	28.433	0.951	0.244	-	1.193

Table 2: The timing results. The timing (in seconds) of SP is the total time for all pages of all participants. The timing (in seconds) of MP is the total time for all pages.

of clusters n_c is chosen as $n_c = \sqrt{N/2}$, where N is the number of participants. Based on the clustering, we allow users to select two participants for comparison as shown in Figures 8 and 9.

6. Results

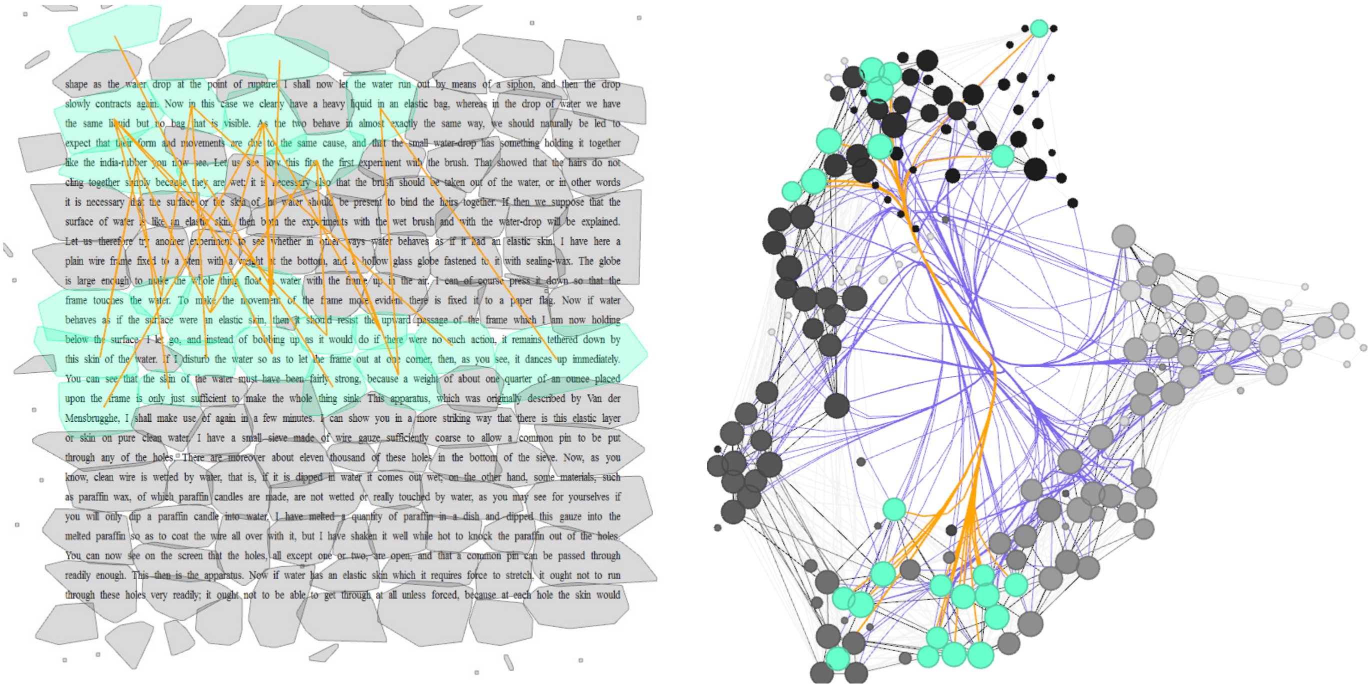
In the section, we first describe the data set and report the timing performance. Then we demonstrate the results for SPSP, SPMP, and MPSP, as well as the benefits and knowledge gained using ETGraph. For the 10 research questions (**Q1** ~ **Q10**) grouped into six categories (**C1** ~ **C6**), we add a note such as (**C1-Q1**) to show the category and the question that each result answers.

6.1. *Data Set and Timing Performance*

The data set was generated from eye-gaze data collected while participants read an excerpt from a book entitled "Soap-bubble and the Forces which Mould Them" [31]. This text was chosen as it was on a novel topic which would be relatively unfamiliar to a majority of readers. Eye-gaze data was collected with a Tobii TX300 remote eye tracker with the sampling frequency of 300 Hz. The eye tracker was affixed below a monitor set to a resolution of 1920×1080, which displayed the text. The excerpt consisted of text from the first 35 pages of the book and contained around 5700 words across 10 pages. Each participant read the text for 20 minutes, and not every one was able to finish reading all pages. Few participants read the final page (Page 10), so it has not been included in our analysis. Eye-gaze data were collected from both eyes and the data from each eye were filtered and averaged together prior to eye-movement detection. The data were then converted into a series of fixations using a dispersion-based filter. Using the Open Gaze And Mouse Analyzer (OGAMA) [32], the filter was set to detect fixations if there were consecutive gaze points within a range of 57 pixels (approximately 1 degree of visual angle) for longer than 100 ms, which is the shortest duration for naturalistic eye-movements during reading [11, 33]. Saccades were then calculated from the fixations. The timing was collected on a PC with an Intel 3.6 GHz CPU and 32 GB memory. The processed data set consists of 27 participants and 9 pages. The timing results are shown in Table 2.

6.2. *SPSP*

Figure 5 shows an example of SPSP. In (a), each dot in the page view represents a fixation and the fixation clusters are highlighted using convex hulls. In (b), each node in the graph view represents a fixation cluster and an edge represents a transition between two clusters. In the graph, the nodes with



(a) (b)

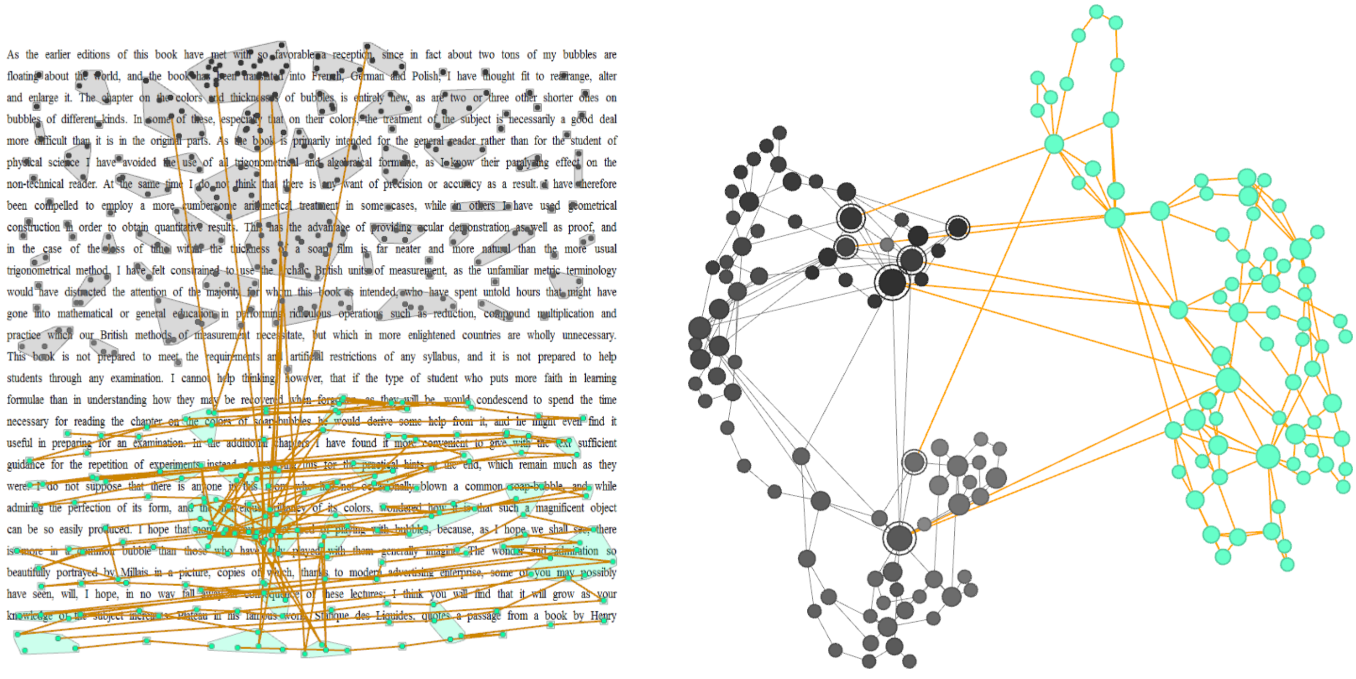
Figure 4: Edge bundling for MPSP. (a) The saccades of the selected bundles illustrate a close relationship between the two green areas in the page view. (b) The bundled edges are highlighted in blue while user-selected edge bundles are shown in orange. The regular (non-outlier) edges are shown in gray with higher frequency edges shown in darker gray.

stronger transitions are placed closer to each other. The graph view provides an overview of the reading pattern for a single page. The darkness of nodes (from dark to light) indicates the presumed reading order. Therefore, in general, nodes with similar darkness values are placed nearby. We can observe that the nodes in (b) are clearly separated into two groups. The nodes in one group are in green and their corresponding fixations are located in the lower portion of the page as shown in (a). The separation between these two groups of nodes could indicate that this participant read the portion of text at the top of the page separately from the portion of text at the bottom of the page. Although some saccade outliers connect the top and bottom portions of the page, more connections exist within the two portions. This indicates that there are stronger connections within each portion than between the two portions. Of particular interest are the nodes in (b) that are not selected but are connected to nodes that are. The corresponding fixations are located in the first few lines of the page, which could indicate that the participant always returned to this location of the page to reread (**C1**-**Q1**).

To understand the scanpath structures, besides regular reading patterns, it is important to analyze saccade outliers. They could indicate a portion of text that is difficult to understand or a rereading pattern. We show an example in Figure 1. In (a) and (b), the fixation clusters and nodes of the saccade outliers are in yellow. The selected clusters and nodes are in green. Red and blue edges indicate backward and forward saccade outliers, respectively. Most of the saccade outliers in (a) are either targeted at or moving from the upper portion of text. This could indicate that the participant reread this portion of text. In addition, there is a saccade outlier from the middle of the page toward the bottom of the page (outside of the screen), but the connected nodes of the saccade outlier are close in (b), which demonstrates that ETGraph gathers the nodes based on their transition relations instead of their spatial closeness. In the time view of (c), the vertical lines indicates the time slots where saccade outliers occurred. The saccade outliers are displayed in red and blue. (d) shows the distribution of the saccade outliers in each of the three sections of the page. This distribution shows that this participant had a large number of saccade outliers in the first section of the page (**C2**-**Q3**).

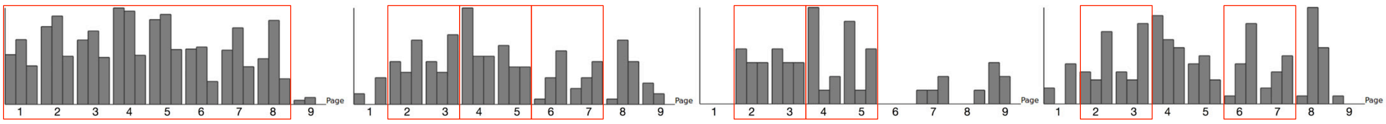
6.3. SPMP

An analysis of the reading patterns of a participant for multiple pages can be done by studying the SPMP-supergraph of the chosen pages along with their statistical information. Figure 3 displays a SPMP-supergraph of nine pages for a single participant. The transition graphs of Pages 1, 5 and 8 are quite simple since each transition graph forms a smooth curve. The graph of Page 3 is very interesting because it consists of two paths from the beginning to the end, which could mean that this participant read the page twice. However, this graph structure is still simple compared to the graphs for Pages 4, 6 and 7. These graphs consist of complex relationships between nodes which could indicate that the participant read these pages backward and forward many times (**C1**-**Q6**). Shifting to the corresponding SPSP view allows for a more detailed examination of these pages, which could offer additional insights.



(a) (b)

Figure 5: The transition graph of SPSP shows the clear separation of two parts in reading by the participant. An interesting repeated rereading pattern is also identified in green. The lower portion of the text in (a) corresponds to the selected nodes in (b).



(a) (b) (c) (d)

Figure 6: Frequency distributions of (a) fixations, (b) saccade outliers, (c) saccade outliers with large *y* distances, and (d) saccade outliers with large *x* distances.

SPMP also allows a comparison of pages by displaying statistical information for each page, which could indicate consistency of reading patterns across all pages. The charts in Figure 6 show the distribution of fixations (a), saccade outliers (b), saccade outliers with large *y* distances (c), and saccade outliers with large *x* distances (d). In (a), the fixation distributions are quite similar among the first eight pages, which indicates similar reading patterns. In (b), (c) and (d), the saccade outlier distributions are similar between Page 2 and Page 3. In (b) and (c), the saccade outlier distributions are similar between Page 4 and Page 5. In (b) and (d), the saccade outlier distributions are similar between Page 6 and Page 7. We conclude that there were similar reading patterns between these pages, especially adjacent pages (**C6**-**Q7**).

Besides showing the structures of the scanpaths, ETGraph may be useful to find the patterns of MWs. Figure 7 shows MWs of a participant. In (a), the pink nodes indicate the appearance of MWs. (b) and (c) show the page and graph views of Page 2, respectively, (d) and (e) show the page and graph views of Page 5, respectively. In (b) and (d), the scanpaths around the MWs consist of saccade outliers. So, in (c) and (e), the subgraphs around the MWs span a large area. We can see that ETGraph could provide a hint for users to detect pages consisting of MWs. If there is more than one episode of MW per page, the subgraphs of MWs may consist of saccade outliers and have overlap between them (**C4**-**Q5**).

6.4. *MPSP*

To show the overall reading patterns of MPSP, we bundle the edges to observe their trends, as shown in Figure 4. In (b), only the saccade outliers are bundled since bundling all edges would hide the trend in the saccade outliers. The regular (non-outlier) edges are shown in various gray colors and the darkness of each edge shows its transition frequency. These gray edges give an impression of common reading patterns (**C1**-**Q8**). In contrast, the saccade outliers are bundled and highlighted in blue. Nodes or bundles can be selected for observing their corresponding fixations and saccades. The selected bundles are shown in orange and their connected nodes are shown in green. The corresponding page view in (a) illustrates a strong relationship between the areas highlighted in green.

Besides showing the overall patterns of all participants, we can also cluster participants based on their attributes, e.g., fixation distribution, graph similarities, and repeated scanpaths (**C5**-**Q10**). Figure 8 shows a comparison of two participants who are in the same cluster based on fixation distribution. In (b), blue nodes belong to one participant and red nodes belong to another, while gray nodes belong to both of them. The dark

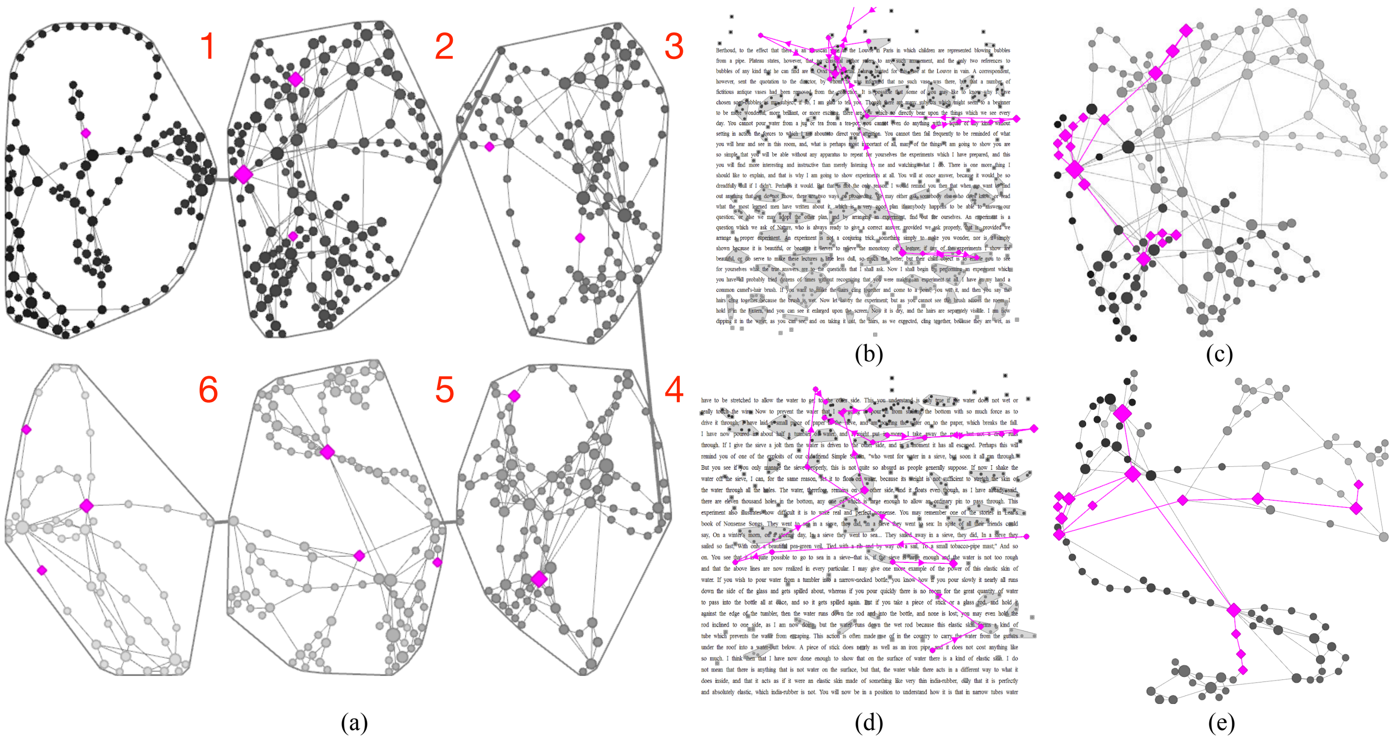


Figure 7: Visualization of MW of a participant for SPMP. In (a), the pink nodes indicate the occurrence of MW. (b) and (c) are the page and graph views of Page 2, respectively. (d) and (e) are the page and graph views of Page 5, respectively.

edges belong to both participants and the gray ones do not. In this example, most of the nodes and a large number of edges are shared, which indicates that their corresponding graphs are also quite similar. From (a), we can see that the two participants share quite a few fixation clusters shown in gray. The clusters that differ are located at the boundaries of the page, shown in blue and red. In addition, they both have a relative lack of fixations in the middle area, highlighted in yellow. (c) displays the time of saccade outliers in the gray time views at the top and bottom, and the shared repeated scanpaths in the middle time view. There are a large number of colored bars which shows that the two participants shared a large number of repeated scanpaths. This further indicates that the two participants had similar reading patterns. (d) compares the fixation distributions of all pages and similar fixation distributions can be observed (**C6-Q2**, **C6-Q9**).

Figure 9 shows a comparison of two participants who are in two different clusters based on fixation distribution. In (b), there are more blue nodes than red nodes. From (a), we can see that there is a large number of blue fixation clusters in the middle of the page and there are no red fixation clusters, which indicates that the participant in red did not read those portions at all. In (c) there are only four colored bars which show that the two participants only shared two repeated scanpaths. This further indicates that the two participants had different reading patterns. (d) compares the statistics of fixations of all the pages and shows different distributions (**C6-Q2**, **C6-Q9**).

We also provide statistical information to help users identify similarities and differences between all participants. For example, in our data, we found that the saccade outliers with large y distances do not occur in any repeated scanpaths for all participants. This indicates that the participants did not revisit two portions of text that have a large y distance. However, it is not the case for saccade outliers with large x distances. Figure 10 (a) shows the repeated scanpaths and saccade outliers with large x distances. The repeated scanpaths are shown in gray while the saccade outliers are highlighted in red and blue bars indicating backward and forward saccade outliers, respectively. Repeated scanpaths occur when a participant rereads some portion of text. A repeated scanpath may represent a scan of the text or a return to a particular sentence (**C3-Q4**). When a participant tries to understand a large paragraph and a repeated scanpath is only a part of it, then this repeated scanpath is only a scan. This is different from when the time gap between the corresponding scanpaths of a repeated scanpath is short and there is a saccade outlier between them. In Figure 10, we can see from (a) that there are some repeated scanpaths overlapped with saccade outliers with large x distances highlighted in the red rectangle. These repeated scanpaths may be more interesting than others. In this figure, repeated scanpaths that consist of saccade outliers are shown in the red rectangle. The corresponding time view is shown in (b), and the corresponding selected repeated scanpaths are shown in (c), (d), and (e). The page view shows that this participant reread this sentence twice, ostensibly for better understanding.

7. User Study and Expert Evaluation

To evaluate the effectiveness of ETGraph, we conducted a user study and an expert evaluation. The user study mainly focused on the usefulness and usability of ETGraph, while the

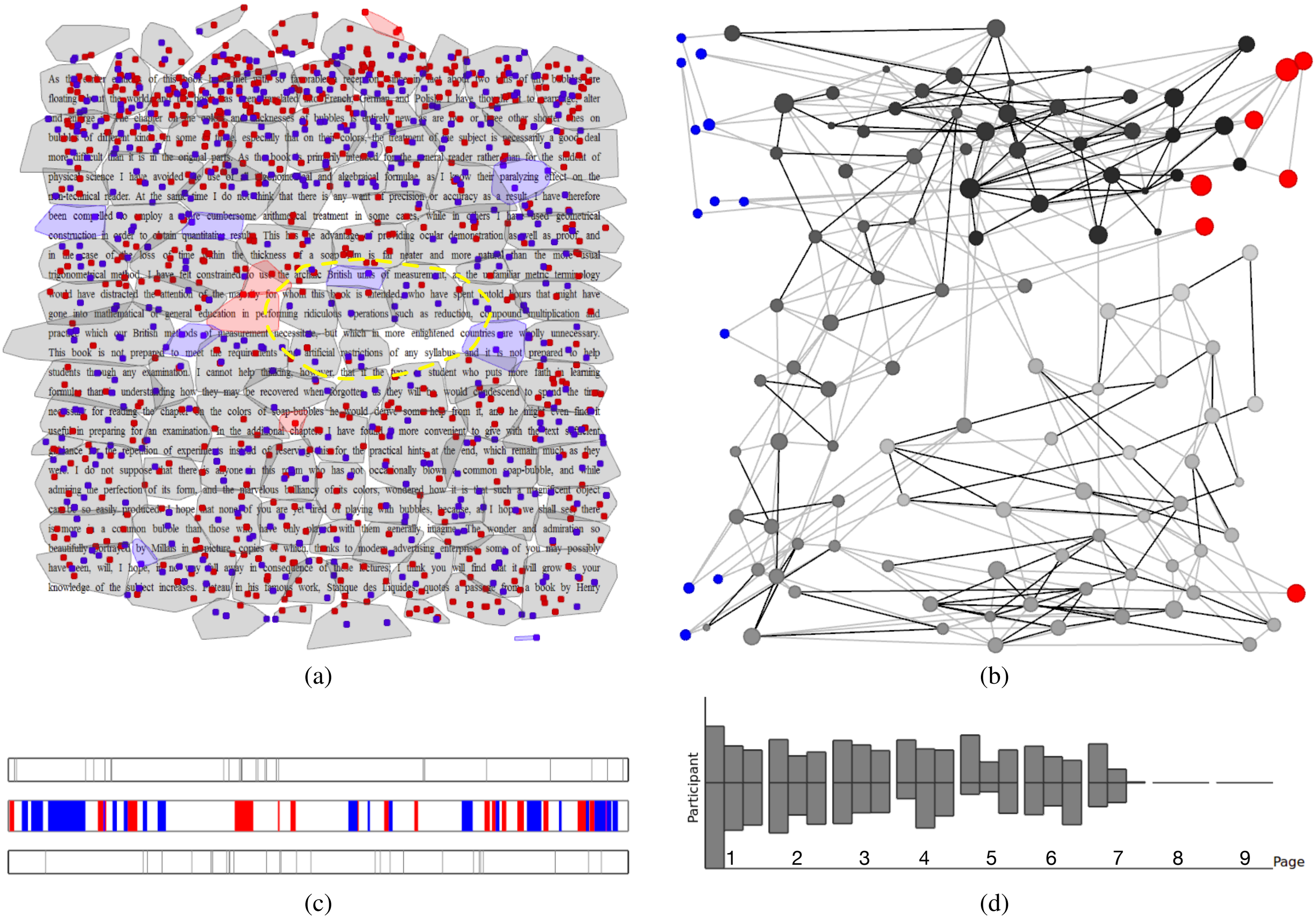


Figure 8: Comparison of two participants with similar fixation distributions. (a) The fixation distributions of the two participants. Note that both participants have few fixations in the middle area (highlighted in yellow). (b) Blue and red nodes belong to a single participant only and gray nodes belong to both participants. Dark edges belong to both participants. (c) Saccade outliers for the two participants are shown in the upper and bottom time views, while their shared repeated scanpaths are shown in the middle view. (d) A comparison of the statistics of fixations of the three sections of each page for the two participants. The participant with red nodes is at the top while the participant with blue nodes is at the bottom.

expert evaluation focused on the improvement and future directions.

7.1. *User Study*

We recruited five unpaid PhD students in our university to evaluate the effectiveness of ETGraph. One student is from the Department of Psychology and four students are from the Department of Computer Science and Engineering. All five users are analyzing MW in their respective PhD studies using different kinds of data, e.g., physiology, facial features, or eye gaze. The user study was conducted in a lab using the same PC for each user. The users were first introduced to ETGraph and were instructed about its design goals and main functions. Then they were given ten minutes for free exploration to get familiar with the system. After that, they were asked to complete six tasks and a survey of seven general questions on the design of ETGraph. These tasks were written on paper and the users hand wrote their responses. The observer (i.e., one author of this work) stood near by and took notes. After the study, he then interviewed the users about their thoughts of the tasks and ETGraph.

Since ETGraph was mainly designed to visualize the reading patterns of the participants and help users identify MWs, this user study focuses on evaluating the effectiveness of these two aspects. As shown in Table 3, **T1** ~ **T3** were designed to evaluate the effectiveness in terms of revealing the reading patterns based on the graph representation, and **T4** ~ **T6** were designed to evaluate the effectiveness in terms of helping the users identify MWs.

In **T1**, the user was given three graphs. She was asked to compare them based on the graph structures and identify the one whose structure is different from those of the other two. Then the user was asked to observe the corresponding scanpaths and understand the relationships between graph structures and scanpaths. Finally, she was asked to verify her observation through freely exploring the graphs and scanpaths. This task was designed to evaluate if the user was able to identify the graph with an abnormal scanpath and infer why the graph structures are different.

In **T2**, given a graph, the user was asked to circle node clusters. Then she was asked to observe their correspondences in

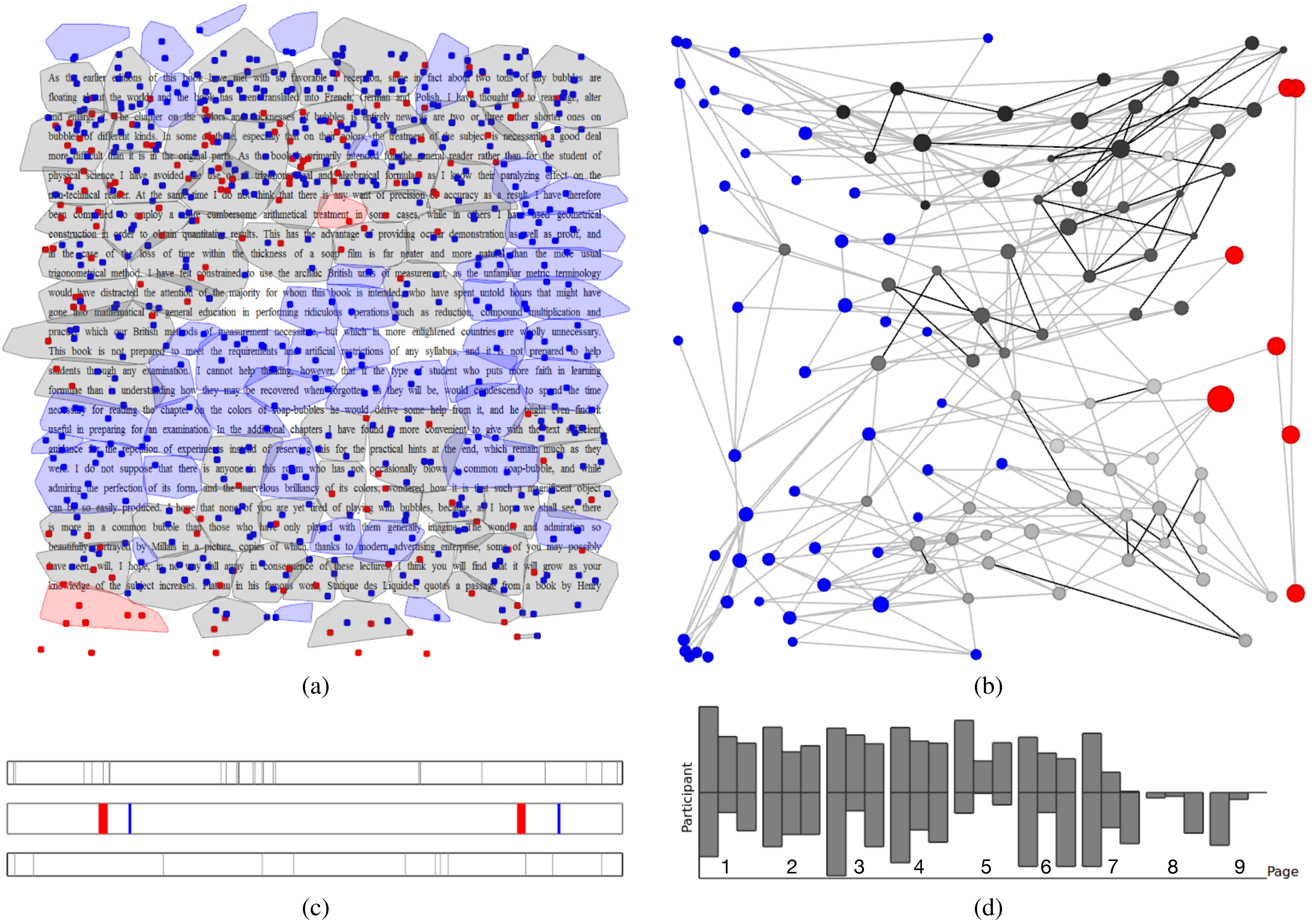


Figure 9: Comparison of two participants with different fixation distributions. (a) The fixation distributions of the two participants. Notice that the participant in red have no fixations in the middle area. (b) Blue and red nodes belong to a single participant only and gray nodes belong to both participants. Dark edges belong to both participants. (c) Saccade outliers for the two participants are shown in the upper and bottom time views, while their shared repeated scanpaths are shown in the middle view. (d) A comparison of the statistics of fixations of the three sections of each page for the two participants. The participant with red nodes is at the top while the participant with blue nodes is at the bottom.

the page view. Finally, she was allowed to explore the other graphs and infer why some graphs had clusters while others did not. This task was designed to evaluate if the user understood that the nodes are placed nearby in the graph view due to their strong relations in the scanpath.

In **T3**, given a graph, the user was asked to estimate which edges in the graph represent saccade outliers. Then she was asked to verify her guesses using ETGraph. Finally, she was allowed to explore the other graphs to observe the correspondence of outliers between the graph view and the page view. This task was designed to evaluate if the user understood that saccade outliers with large y distances exist between two nodes with either very different gray-scale colors or very long edges.

The typical way for MW detection is to follow the animation of scanpath and identify MWs based on user experience. However, the scanpath could be long, dense, and self-occluded, which makes it difficult for users to follow the animation, memorize the animation history, and detect abnormal reading patterns. ETGraph simplifies the scanpath during animation by preserving the most important features and reducing the user's effort. Furthermore, since identifying MWs by watching the animation requires a lot of domain knowledge, ETGraph simplifies the process by helping users detect MWs through showing a static view of the graph structures and visual hints for saccade outliers. **T4** ~ **T6** were designed to evaluate the effectiveness of ETGraph in terms of helping identifying MWs. In **T4**, the user was asked to watch the scanpath animation of several participants reading different pages, and identify whether MW was reported on each page. **T5** and **T6** asked the user to identify whether MW was reported on the pages based on the graph structures and saccade outliers, respectively.

The users could perform the tasks at their own pace. Each session took about 30 to 60 minutes to complete. We summarize the binary task completion scores for the six tasks in Table 3.

We note that all users answered **T1** and **T2** correctly. For **T1**, the users noticed that when a participant read from left to right and from top to bottom, then the corresponding graph presents a continuous transition from darker nodes to lighter nodes, which conforms to the presumed reading order. How-

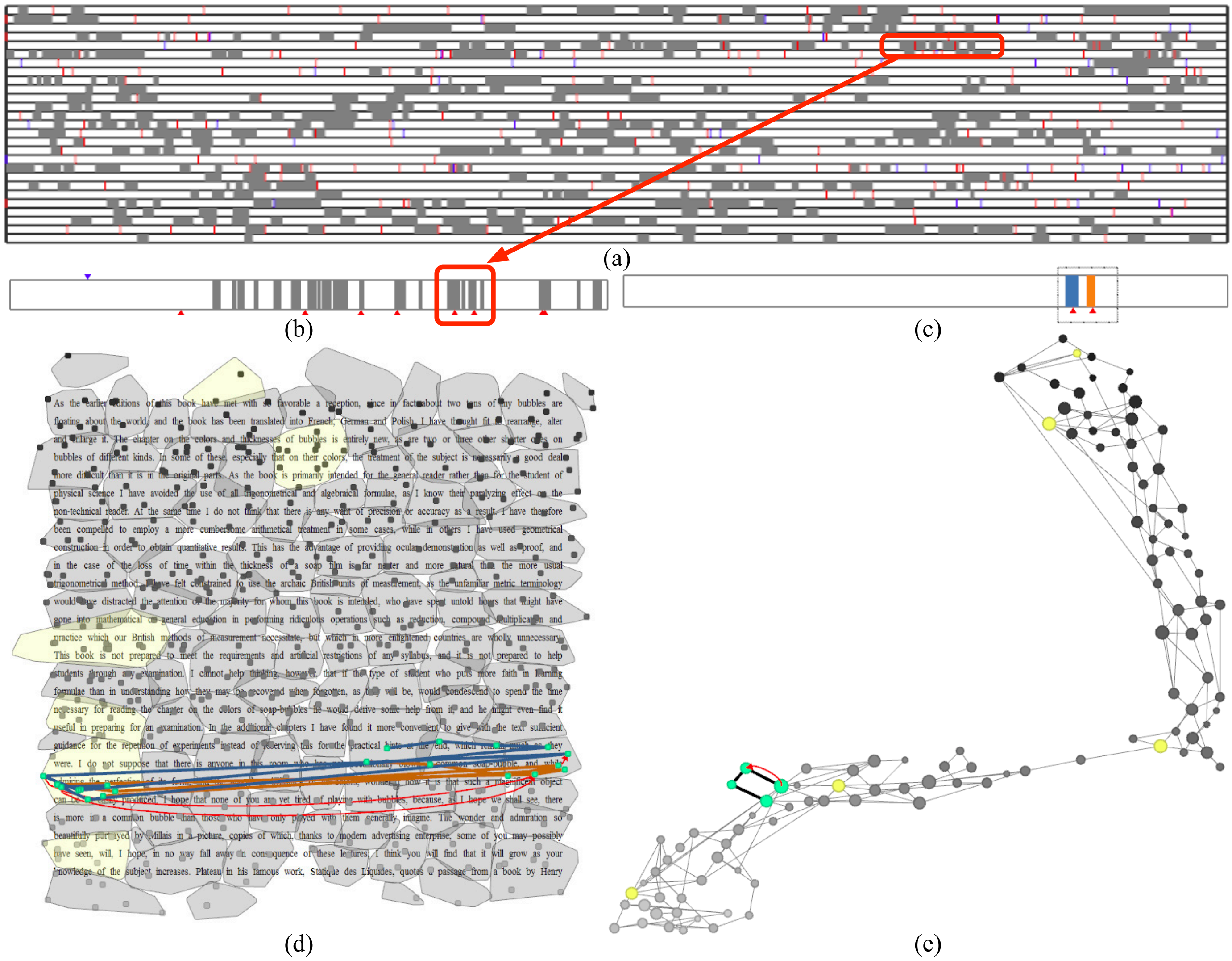


Figure 10: (a) The time view of saccade outliers and repeated scanpaths for MPSP, organized in a scarf plot. Repeated scanpaths are shown in gray while saccade outliers are highlighted in red and blue bars, indicating backward and forward saccade outliers, respectively. The red rectangle marks the repeated scanpaths that overlap with saccade outliers. (b) A participant is selected from (a) and the corresponding time view is shown. (c) The repeated scanpaths are selected. (d) and (e) are the corresponding page view and graph view for the selected repeated scanpaths, respectively.

ever, the graph structure became complex when there was a large number of saccade outliers. For **T2**, the users drew circles to highlight the clusters in the graph, and selected each group to verify their estimates. They concluded that if the participants separated the text into portions and read each portion carefully, then each portion would form a cluster in the graph.

For **T3**, the users could identify most of the saccade outliers with large y distances based on the edge lengths of the graph, but there was one saccade outlier that they all failed to identify. The edge of that saccade outlier linked two nodes that were close in the graph but one node was very dark and the other was very light. The fixation of the light node was very close to the bottom of the page and the dark node consisted of a lot of fixations at the top of the page. Therefore, the dark node pulled the light one close to itself and the corresponding edge length was small. Failing to identify such an outlier shows that we should have explained the details of ETGraph construction and the layout generation algorithm to users. This would help them better understand the graph so they would know that the edge represents a saccade outlier when they observe such a phenomenon.

For **T4**, the users tended to animate the entire scanpath to understand the reading pattern and identify possible MW. However, it was sometimes difficult for them to do this because of visual clutter and the effort of remembering the previous scanpath. To help users keep track of the reading pattern, ETGraph only displays a few of the most currently displayed saccades and all the previous saccade outliers during the animation. The average score of successfully identifying MW was 0.7 for this task and most of the users considered this task difficult to complete. All users except one made at least one incorrect judgement. Most of the users stated that viewing an animation based on time that included saccade outliers helped them identify rereading behaviors and abnormal reading patterns that may include MW, but this function still requires them to have some

task	description	average score	standard deviation
T1	Identify the abnormal reading pattern based on the graph representations, and study their corresponding scanpaths.	1	0
T2	Circle the clusters in the graph and observe their corresponding portions in the page view.	1	0
T3	Circle the saccade outliers with large y distances in the graph view and verify your guesses in the page view.	0.8	0
T4	Identity the pages with MW based on the animation of the whole scanpath.	0.7	0.21
T5	Identify the pages with MW based on the graph structures.	0.8	0.27
T6	Identify the pages with MW based on the saccade outliers in the graph view.	0.8	0.45

Table 3: The six tasks in the user study and user scores of the tasks.

knowledge about typical reading patterns associated with MW.

For **T5**, the average score of successfully identifying MW was 0.8. All users stated that the graph structure could help determine whether the participant read normally or was MW. This task consisted of two subtasks. Three users who considered this task easy completed it correctly. The remaining two users only answered one small part of the task correctly. They both agreed that Figure 7 (c) (used in **T5**) showed a graph with a normal structure. This result indicates that we need to train users about the graph structure in order to distinguish the differences between graphs with and without MW.

For **T6**, the average score of successfully identifying MW was 0.8. This task consisted of two subtasks. Four users who considered this task easy completed it correctly. The remaining user did not respond correctly. He studied the content connected by the saccade and decided that there was no MW present if the content was related. However, this is not necessarily always the case.

On the general questions, all users agreed that ETGraph was helpful for studying eye-tracking data. Based on the graph structure, they could get an impression on whether the participant followed a regular reading pattern or not. Saccade outlier detection helped them identify saccade outliers and possible MW. Users also had some suggestions to improve ETGraph. Four of them suggested that we should provide more training and explanation of ETGraph for better use. One even suggested that we list pages with or without MW, so that their differences would be more obvious in ETGraph. Two users suggested that we develop an iPad or Windows version for them to explore further as our current system runs on Linux.

7.2. Expert Evaluation

We also invited two domain experts: a professor and a PhD student whose research interest is identifying MW in eye-tracking data. We utilized the think-aloud protocol during the evaluation. The experts described their thoughts while completing the tasks, and we summarized their comments after the evaluation. They both agreed that ETGraph is a very helpful tool for researchers who are interested in studying eye movements.

The professor considered ETGraph a useful tool to identify the reading patterns around MWs. He pointed out some suggestions to improve ETGraph. First, he suggested that screen space should be added for the graph view so that users can select multiple graphs for comparison. Second, he suggested that ETGraph should focus more on saccade outliers with large y distances because they are important to identify rereading and MW. Saccade outliers with large x distances might be due to the poor calibration of eye trackers and line jumps. For MPSP, he thought bundling outliers was helpful to identify the trend of abnormal reading patterns. However, he suggested that we bundle the outliers separately for pages with or without MW. This could help researchers study common patterns of MWs. Finally, he thought that after identifying the common patterns of the graphs with MW, using graph similarity measures could help identify the pages with MW.

The student expert considered ETGraph to be a useful tool because it provides a new approach to visualize eye-tracking data. He has used tools like the Open Gaze and Mouse Analyzer (OGAMA) [32] and has even written his own programs to analyze eye movements. These tools render the data using a scanpath or heatmap. Visual clutter is inevitable when displaying a scanpath, while heatmaps lack saccade information. ETGraph addresses both limitations. In addition, he thought that ETGraph helps to not only visualize eye movements but also gain an understanding of patterns that are not obvious with other tools. In addition, he is particularly interested in the repeated scanpath detection function. For him, this function provides additional information drawn from ETGraph besides saccade outliers that could be used to detect MW. Finally, he thought that our approach could help to engineer novel features for use in machine learning based on observations drawn from ETGraph.

8. Conclusions and Future Work

We have presented ETGraph, a new approach that transforms eye-tracking data of reading studies from the original page view to a graph-based representation. The graph view presents fixation clusters as nodes and saccades as edges, and it can reveal the very nature of the reading pattern by placing nodes in the graph according to their connections, rather than their fixed locations in the page. Through brushing and linking, users are able to explore eye-tracking data from multiple perspectives. We demonstrate the usefulness of ETGraph by presenting results generated from studying single participant single page, single participant multiple pages, and multiple participants single page. The feedback from the domain experts and a group of student researchers confirms the effectiveness of our approach.

The current implementation of ETGraph has some limitations. First, ETGraph helps users identify whether or not MWs happen in pages, but it cannot tell accurately when and where MWs happen. Second, MW detection in ETGraph may not be extended to analyzing other types of data such as videos. MW detection in our reading context assumes that the participants should follow a normal reading pattern (from left to right, from top to bottom), and if they do not, MW may happen. Clearly, this assumption does not hold anymore for other types of data.

Third, the clustering of participants is based on their graph similarities, which does not allow users to manually select certain graph attributes for more flexible participant classification.

In the future, we would like to further apply graph mining techniques, such as solutions for graph alignment or matching, to ETGraph, and investigate common reading patterns and abnormal reading behaviors. The goal is to help users detect a wide variety of cognitive and affective phenomena, such as mind wandering, cognitive load, inference generation, and boredom, in a visually guided manner. We would also develop graph-based visual analytics tools for studying some other eye-tracking data, such as data recorded for dynamic stimuli such as videos.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation (NSF) through grants DRL-1235958, IIS-1456763, IIS-1455886, and IIS-1523091. Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF. The authors would like to thank the anonymous reviewers for their insightful comments.

References

[1] Smallwood J, Schooler JW. The restless mind. Psychological Bulletin 2006;132(6):946–58.
[2] Killingsworth MA, Gilbert DT. A wandering mind is an unhappy mind. Science 2010;330(6006):932–.
[3] Smallwood J, Fishman DJ, Schooler JW. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. Psychonomic Bulletin & Review 2007;14(2):230–6.
[4] Kane MJ, McVay JC. What mind wandering reveals about executive-control abilities and failures. Current Directions in Psychological Science 2012;21(5):348–54.
[5] Randall JG, Oswald FL, Beier ME. Mind-wandering, cognition, and performance: A theory-driven meta-analysis of attention regulation. Psychological Bulletin 2014;140(6):1411–31.
[6] Bixler R, D'Mello S. Toward fully automated person-independent detection of mind wandering. In: Proceedings of International Conference on User Modeling, Adaptation, and Personalization. 2014, p. 37–48.
[7] Reichle ED, Reineberg AE, Schooler JW. Eye movements during mindless reading. Psychological Science 2010;21(9):1300–10.
[8] Feng S, D'Mello S, Graesser A. Mindwandering while reading easy and difficult texts. Psychonomic Bulletin & Review 2013;20(1):586–92.
[9] Smallwood J, Davies JB, Heim D, Finnigan F, Sudberry M, O'Connor R, et al. Subjective experience and the attentional lapse: Task engagement and disengagement during sustained attention. Consciousness and Cognition 2004;13(4):657–90.
[10] Franklin MS, Broadway JM, Mrazek MD, Smallwood J, Schooler JW. Window to the wandering mind: Pupillometry of spontaneous thought while reading. The Quarterly Journal of Experimental Psychology 2013;66(12):2289–94.
[11] Rayner K. Eye movements in reading and information processing: 20 years of research. Psychological Bulletin 1998;124(3):372–422.
[12] Franklin MS, Smallwood J, Schooler JW. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. Psychonomic Bulletin & Review 2011;18(5):992–7.
[13] Bixler R, D'Mello S. Toward fully automated person-independent detection of mind wandering. In: Proceedings of International Conference on User Modeling, Adaptation, and Personalization. 2014, p. 37–48.
[14] Raschke M, Blascheck T, Burch M. Visual analysis of eye tracking data. In: Huang W, editor. Handbook of Human Centric Visualization. Springer New York; 2014, p. 391–409.
[15] Tory M, Atkins MS, Kirkpatrick AE, Nicolaou M, Yang GZ. Eyegaze analysis of displays with combined 2D and 3D views. In: Proceedings of IEEE Symposium on Information Visualization. 2005, p. 519–26.
[16] Duchowski AT. A breadth-first survey of eye tracking applications. Behavior Research Methods, Instruments, and Computers 2002;34(4):455–70.
[17] Blascheck T, Kurzhals K, Raschke M, Burch M, Weiskopf D, Ertl T. State-of-the-art of visualization for eye tracking data. In: Eurographics Conference on Visualization - State-of-the-Art Reports. 2014, p. 83–2.
[18] Santella A, DeCarlo D. Robust clustering of eye movement recordings for quantification of visual interest. In: Proceedings of Symposium on Eye Tracking Research and Applications. 2004, p. 27–34.
[19] Špakov O, Räihä KJ. KiEV: A tool for visualization of reading and writing processes in translation of text. In: Proceedings of Symposium on Eye Tracking Research and Applications. 2008, p. 107–10.
[20] Goldberg JH, Helfman JI. Scanpath clustering and aggregation. In: Proceedings of Symposium on Eye Tracking Research and Applications. 2010, p. 227–34.
[21] Tang S, Ronan GR, Christian V. EyeMap: A software system for visualizing and analyzing eye movement data in reading. Behavior Research Methods 2012;44(2):420–38.
[22] Tsang HY, Tory M, Swindells C. eSeeTrack - visualizing sequential fixation patterns. IEEE Transactions on Visualization and Computer Graphics 2010;16(6):953–62.
[23] Burch M, Beck F, Raschke M, Blascheck T, Weiskopf D. A dynamic graph visualization perspective on eye movement data. In: Proceedings of Symposium on Eye Tracking Research and Applications. 2014, p. 151–8.
[24] Burch M, Kull A, Weiskopf D. AOI Rivers for visualizing dynamic eye gaze frequencies. Computer Graphics Forum 2013;32(3):281–90.
[25] Andrienko G, Andrienko N, Burch M, Weiskopf D. Visual analytics methodology for eye movement studies. IEEE Transactions on Visualization and Computer Graphics 2012;18(12):2889–98.
[26] Blascheck T, John M, Kurzhals K, Koch S, Ertl T. VA^2: A visual analytics approach for evaluating visual analytics applications. IEEE Transactions on Visualization and Computer Graphics 2016;22(1):61–70.
[27] Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. Software - Practice and Experience 1991;21(11):1129–64.
[28] Gu Y, Wang C. TransGraph: Hierarchical exploration of transition relationships in time-varying volumetric data. IEEE Transactions on Visualization and Computer Graphics 2011;17(12):2015–24.
[29] Ukkonen E. On-line construction of suffix trees. Algorithmica 1995;14(3):249–60.
[30] Holten D, van Wijk JJ. Force-directed edge bundling for graph visualization. Computer Graphics Forum 2009;28(3):983–90.
[31] Boys CV. Soap-Bubbles, and the Forces which Mould Them. Cornell University Library; 1890.
[32] Voßkühler A, Nordmeier V, Kuchinke L, Jacobs AM. OGAMA (open gaze and mouse analyzer): Open-source software designed to analyze eye and mouse movements in slideshow study designs. Behavior Research Methods 2008;40(4):1150–62.
[33] Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, Van de Weijer J. Eye Tracking: A Comprehensive Guide to Methods and Measures. Oxford University Press; 2011.