COMPUTER
ENGINEERING
PROGRAM

Middle East Technical
University
Northern Cyprus
Campus

eMINE Technical Report Deliverable 5 (D5),

September 2013

eMINE
Scanpath Analysis Algorithm

Sukru Eraslan [1,2], **Yeliz Yesilada** [1] **and Simon Harper** [2]
[1] Middle East Technical University,
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, TURKEY
{seraslan,yyeliz}@metu.edu.tr

[2] School of Computer Science,
The University of Manchester,
Oxford Road, Manchester, UK,
M13 9PL
sukru.eraslan@postgrad.manchester.ac.uk
simon.harper@manchester.ac.uk

The existing re-engineering, namely transcoding, techniques improved disabled and mobile Web users experience by making Web pages more accessible in constrained environments such as on small screen devices and in audio presentation. However, none of these techniques use eye tracking data to transcode Web pages based on understanding and predicting users' experience. The overarching goal of eMINE project is to improve the user experience in such constrained environments by using a novel application of eye tracking technology. The project proposes an algorithm to identify common scanpaths, which are eye movement sequences, and relating those scanpaths to visual elements of Web pages. It can then be used to transcode Web pages, for instance, unnecessary information can be removed and/or the visual elements can be re-ordered. We assert that both visually disabled and mobile users would benefit from such development.

METU
NCC

eMINE

The World Wide Web (web) has moved from the Desktop and now is ubiquitous. It can be accessed by a small device while the user is mobile or it can be accessed in audio if the user cannot see the content, for instance visually disabled users who use screen readers. However, since web pages are mainly designed for visual interaction; it is almost impossible to access them in alternative forms. Our overarching goal is to improve the user experience in such constrained environments by using a novel application of eye tracking technology. In brief, by relating scanpaths to the underlying source code of web pages, we aim to transcode web pages such that they are easier to access in constrained environments.

Acknowledgements

The project is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with the grant number 109E251. As such the authors would like to thank to TÜBİTAK for their continued support.

Contents

Middle East Technical University,
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, TURKEY

Tel: +90 (392) 661 2000
http://www.ncc.metu.edu.tr/

Corresponding author:
Sukru Eraslan
Tel: +90 (392) 661 2971
seraslan@metu.edu.tr

1 Introduction

The World Wide Web (Web) can be accessed by different devices with different requirements and constraints. Many people access the Web using their small screen devices while they are mobile and visually disabled people typically access the Web using screen readers [26, 86]. When people access Web pages with their small screen devices, they can experience many difficulties [66]. For example, they may need to scroll or zoom a lot which can be annoying and it can be costly to download complex and long pages [66]. Likewise, Web experience can be challenging for visually disabled users [86]. As screen readers follow the source code of Web pages, visually disabled people have to listen to unnecessary clutter to get to the main content and they can hear some meaningless words on poorly designed Web pages [86].

To address these problems Web page transcoding has been proposed. Transcoding is a technique used to re-engineer Web pages to make them more accessible [5]. Although the existing transcoding techniques improve disabled and mobile Web users' experience on the Web [87, 5], none of these techniques use eye tracking data to re-engineer, namely transcode, Web pages based on understanding and predicting users' experience.

The overarching goal of eMINE project is to improve the user experience in such constrained environments by using a novel application of eye tracking technology. Eye tracking has widely been used to investigate cognitive processes for over 30 years [62], but it is relatively a new area in the Web use [38, 49, 11]. While reading, the eyes make quick movements which are called saccades [57]. Between the saccades, the eyes make fixations where they become relatively stationary [57]. Scanpaths are sequences of fixations and saccades on visual stimulus [57]. Figure 1 shows an example scanpath where larger circles represents longer fixations and lines show saccades.

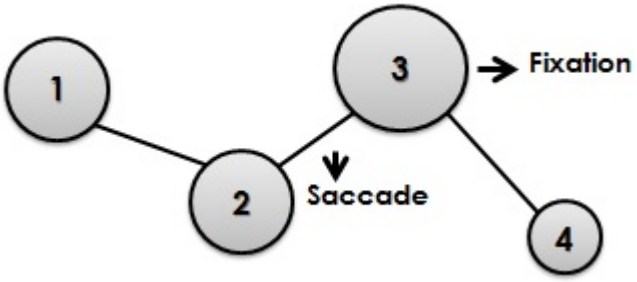


Figure 1: Example scanpath where fixations represent the points fixated by a user and saccades represent quick eye movements between fixations

The **objective** is to use eye tracking data to generate an algorithm for identifying common scanpaths and relating those scanpaths to elements of Web pages, such that Web pages can be transcoded to improve the user experience. In order to achieve the objective, we reviewed the literature for scanpaths analysis methods.

Scanpaths on Web pages have been analysed using different methods and algorithms. Most of them use string representations of scanpaths. String representations are created using the sequence of Areas of Interests (AoIs) that get fixations [75]. Web page AoIs can be generated in different ways such as using a grid-layout [75], the source code of Web

pages [2] or the fixations' distribution over Web pages [68].

The Levenshtein Distance (String-Edit) algorithm has been widely used to analyse scanpaths [38, 75]. This algorithm calculates the dissimilarity between scanpaths by transforming the string representation of one scanpath into another one's string representation using a minimum number of operations which are insertion, deletion and substitution. For instance, the dissimilarity between ABCD and ABCE is calculated as 1 (one) by the String-Edit algorithm because the transformation can be done by only substituting D with E. The dissimilarities can be used to categorise scanpaths [82]. Also, it can be used to investigate differences between the behaviours of people on Web pages [38]. However, this algorithm has considerable weaknesses. Firstly, the substitution costs between all pairs of AoIs may not be the same because their size and distances between AoIs may be different. Substitution cost matrix can be used to store the substitution costs for all pairs of AoIs and then this matrix can be used while calculating the dissimilarity [75]. The Needleman and Wunsch algorithm which uses a substitution cost matrix to calculate the similarity between two strings was also used for scanpath analysis [18]. Secondly, the String-Edit algorithm does not consider duration metrics, such as fixation duration. However, these metrics can be interpreted in different ways. For example, longer fixations can be interpreted as the difficulty for extracting information [57]. ScanMatch method considers fixation duration by defining a particular time duration to cause repetitions of the AoI names in string representations of the scanpaths [16]. For example, if the fixation duration is 200ms in AoI A and the particular duration is 100ms, A is duplicated.

These methods and algorithms have been applied to scanpaths in pair-wise manner. However, this project aims to find common patterns in a group of scanpaths. One of the techniques used is the Transition Matrix which can be created using more than two scanpaths [82]. In this matrix, each cell has the row and column probabilities. Row probabilities allow identifying the next AoI of the particular AoI and column probabilities allow identifying the previous AoI of the particular AoI. When the transition matrix is tried to be used to identify a common scanpath, some significant problems arise: What is the start and the end point of the common scanpath? and Which probabilities should be considered?

To address these problems, some other methods can be considered. The shortest common sub-sequence method has been mentioned in the literature to determine a common scanpath for more than two people but this method has significant weaknesses [60]. For instance, it produces ABDFE as a common scanpath for scanpaths ABE, ADE and AFE. The common scanpath is longer than all of the three scanpaths and it is not supported by the individual scanpaths, for instance, ABE does not include D which is included by the common scanpath.

Some methods, such as T-Pattern [49] and eyePatterns's discover patterns technique [82], try to detect sub-patterns in eye tracking scanpaths. eyePatterns's discover patterns technique have no tolerance to extra items [82]. For instance, ABC can be found as a sub-pattern for ABC and ABCD but it cannot be found for ABC and ABXCD because of X. This example illustrates that this technique is reductionist. eyePatterns also has a method to locally align only two sequences to detect sub-patterns [82]. Furthermore, eSeeTrack tool shows the sequence in a timeline which shows fixation durations [79]. It also visualises the transition between AoIs in hierarchical structure with highlighting the probabilities. For instance, it can show that when people look AoI A, they are likely to look AoI B.

Using multiple sequence alignment was proposed to identify an average scan pattern,

namely a common scanpath, but it was not validated [28]. Also, the Dot-plots based algorithm which constructs a hierarchical structure by finding a common scanpath of two sequences with the Dot-plots algorithm was proposed by Goldberg and Helfman (2010). The scanpaths are leafs and the common scanpath is the root of the hierarchical structure. In order to address the reductionist approach of the Dot-plots algorithm, some statistical methods have been applied.

Common patterns in eye tracking data will allow transcoding Web pages for a wide range of users. If Web pages are transcoded based on a particular scanpath, the transcoded version may not be suitable for many users. Therefore, Web pages will be transcoded based on common patterns. Different ways have been used to transcode Web pages such as adjusting sizes [39], adding a skip link [74], generating summaries of the content [84, 83, 12, 44], generating thumbnail images of Web pages [83, 14], ranking and reordering the content [88, 46], providing a table of contents to reach Web page elements [87], allowing easy navigation between important chunks of Web pages through a consistent set of key presses [44] and removing irrelevant or repetitive content [46, 87].

It is observable that the existing transcoding techniques improved disabled and mobile Web user's experience. These techniques tended to focus underlying source code of Web pages for visual rendering but most of them did not concentrate on understanding and predicting users' experience. However, Web pages can be transcoded more efficiently with good understanding of structure, content and context of use [85].

Eye tracking may allow us to drive Web page transcoding by providing a better understanding of a user's experience and enabling to predict future interactions. Firstly, it should be understood how sighted users read Web pages on desktop screens. Web pages can then be transcoded to improve the user experience. We assert that both visually disabled and mobile users would benefit from such development. Most mobile operators are also interested in transforming Web pages before they are served to end user [1], so the results would be beneficial for mobile operators. Moreover, this project will provide benefits for designers, engineers, and practitioners working on Web accessibility and the mobile Web. Since no existing transcoding techniques consider eye tracking scanpaths, our work will be a practical contribution to Human Computer Interaction and Web Science fields.

The rest of this technical report has the following three main chapters:

Literature Review chapter provides a comprehensive review of eye tracking studies and the existing scanpath analysis methods. After that, it discusses possible scanpath analysis methods which have been applied successfully in other fields, not in eye tracking.

Preliminary Experiments chapter discusses the results of our experiments with the existing and possible scanpath analysis methods. To address the weaknesses of the existing methods we developed eMINE scanpath analysis algorithm.

eMINE Scanpath Analysis Algorithm chapter explains eMINE scanpath analysis algorithm with an initial informative validation.

Summary chapter involves the brief summary of this report. In addition, it states some significant issues to be considered as future work.

[1] http://www.w3.org/TR/ct-guidelines/

2 Literature Review

Scanpath can be defined as an eye movement sequence which shows which points are fixated and which sequence is followed by a user. Figure 2 illustrates a scanpath on a segmented Web page where cycles illustrate fixations and lines represent saccades.

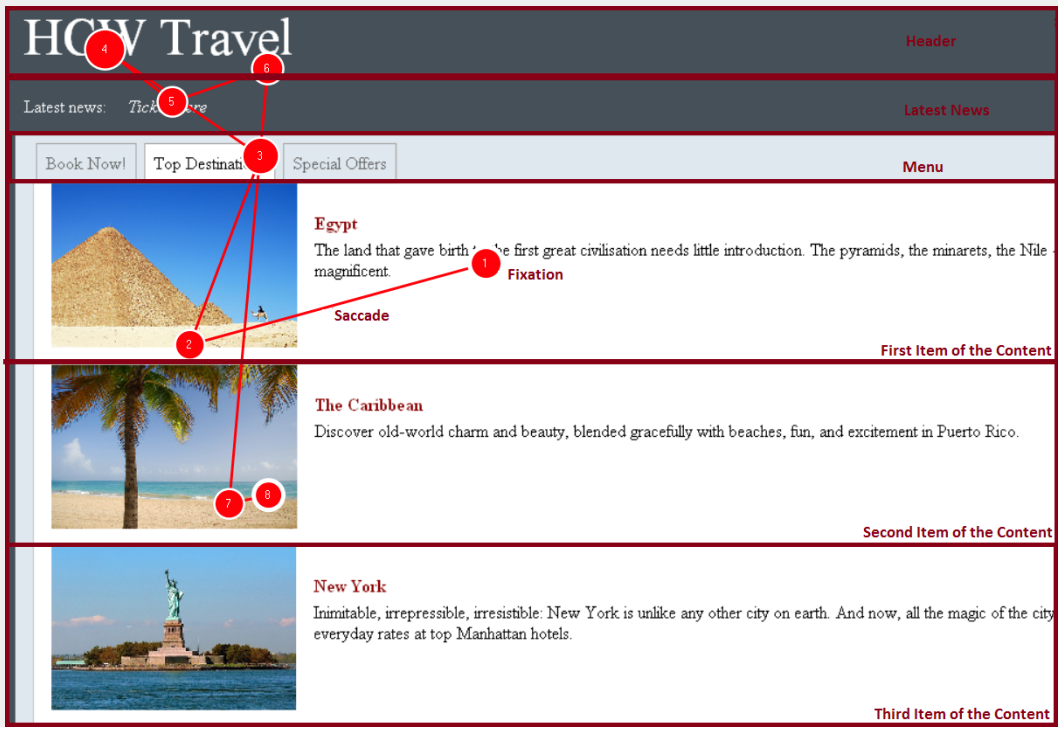


Figure 2: Scanpath on a segmented Web page

This section has a comprehensive review of the eye tracking studies on the Web and scanpath analysis methods. Eye tracking terms used in the literature can be found in the Glossary.

2.1 Eye Tracking Metrics

People may read Web pages by following different paths [41]. Assume that people query a word from a search engine and get the list of search results. Some people prefer to examine each result from the top and then decide immediately whether open it or not whereas some people prefer to look a number of search results and then re-visit one of them to open [41]. There are also some people follow both strategies [41]. It shows that people may follow different paths on Web pages, so it can be difficult to identify a common scanpath on a Web page for multiple people.

Complexity of Web pages causes variances in people's eye movements [55]. Pan et al. (2004) suggests that complex Web pages have larger variances compared to simple Web pages. Figure 3 shows people's scanpaths variance on different Web pages which have different levels of complexity. EBAY-1 and EBAY-F are different pages from EBAY Web site and EBAY-1's structure is simpler than EBAY-F's structure. As shown in the figure,

EBAY-1 has lower variance. Moreover, there is a considerable difference between task completion time in simple and complex pages. Their participants needed 50 % more time to complete the tasks on the complex pages.

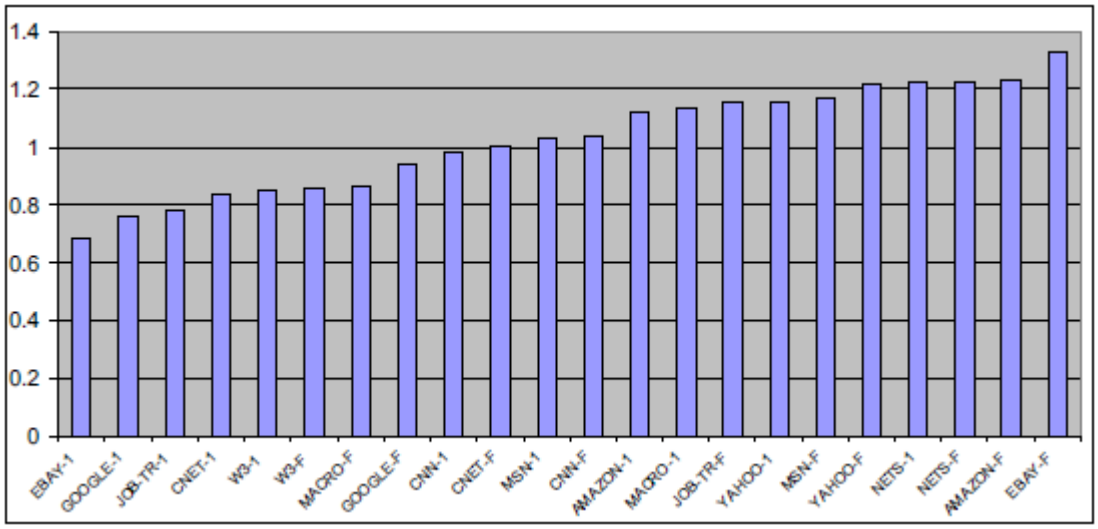


Figure 3: The variance in people's scanpaths on Web pages which have different levels of complexity [55]

Web page familiarity may also affect people's eye movements [51]. In a study conducted by Mccarthy et al. (2003), the participants were asked to complete a number of tasks on the Web pages where a main menu was located at different locations: on the left, on the right and at the top. Since the participants expected to see a main menu on the left, they completed the tasks earlier on the page whose main menu was on the left. After the first trial, the participants became familiar with the Web pages and there was no considerable differences in task completion time.

Eye movements on Web pages can be affected by task difficulty, too [89]. For instance, there can be significant differences in people's eye movements while they are searching noticeable and unnoticeable items.

People's interaction with Web pages can vary because of the reasons explained above. Assume that people are asked to complete some tasks on a particular Web page. Some people complete the tasks easily whereas other people experience some difficulties. Since the commonality decreases, it can be difficult to identify a common scanpath without separating these people. Eye tracking metrics allow identifying these types of differences in data. Table 1 summarises the eye tracking metrics by grouping them into four categories which are duration, frequency, regression and spatial metrics. This table uses ▲to show an increase and ▼to represent a decrease in cause and effect. For example, according to this table, the fixation duration metric is associated with the duration metrics. In addition, when the fixation duration increases, the difficulty of extracting information from the display increases. Since these metrics have widely been mentioned in the literature, they should be considered.

Eye Tracking Metrics					
	Metrics	Cause	Measurement	Effect	Reference
Duration	Fixation Duration	▲	Difficulty	▲	[57]
	Gaze Duration	▲	Difficulty	▲	[35]
	Scanpath Duration	▲	Scanning Efficiency	▼	[24, 57, 20]
Frequency	Number of Saccades	▲	Searching	▲	[24]
	Number of Fixations Overall	▲	Search Efficiency	▼	[24]
	On-target Fixations	▲	Search efficiency	▲	[24]
	Number of Fixations per AoI	▲	AoI Importance	▲	[58]
	Number of Gazes per AoI	▲	AoI Importance	▲	[35]
Regression	Regressive Saccades	▲	Difficulty	▲	[63]
Spatial	Scanpath Length	▲	Search efficiency	▼	[25]
	Saccade Length	▲	Meaningfulness	▲	[25]

Table 1: Summary of the eye tracking metrics where ▲represents an increase and ▼represents a decrease in cause and effect

2.1.1 Duration Metrics: Fixation, Gaze and Scanpath Durations

Fixation duration represents the duration which a fixation last. Fixation duration is proportional to the diameter of the cycle [78]. Therefore, a cycle with a large diameter represents a longer fixation [78]. In Figure 2, the fixation no:4 is the longest fixation. Longer fixations usually indicate the difficulty while extracting information from visual stimuli [35, 24]. Fixation duration can also be affected by gender [55]. One study states the male participants had longer mean fixation duration compared to the female participants' mean fixation duration [55].

Rayner et al. (2009) state that people normally requires at least 150 msec seeing screen in order to process it. Besides this, the short fixations, especially fixations whose durations are less than 100 msec, are discarded by many researchers [73]. For instance, fixations whose durations are below 100 msec have been excluded by [6, 61] which are recent eye tracking studies. However, this approach has a considerable problem. Assume that a person may look AoI A for 50 msec and then look AoI B for 120 msec. Next, s/he looks again AoI A for 90 msec. If the fixations whose durations are less than 100 msec are discarded, AoI A is discarded because it does not involve a fixation whose duration is longer than 100 msec, even though it may be considerable because it is visited twice. Gaze duration becomes critical at this point.

Gaze duration is a sum of the durations of a series of consecutive fixations within a gaze which can be an AoI [35, 57]. Assuming that the second item of the content in Figure 2 is a gaze. Its duration is equal to 220 msec when the fixation no:7 and no:8 durations are 100 msec and 120 msec (100 ms + 120 ms = 220 msec). Similar to fixation duration, this metric indicates the difficulty of extracting information from visual stimuli [35].

Last duration metric is scanpath duration which represents the duration [20]. It is usually equal to the sum of the durations of the fixations and saccades on the scanpath [24]. In particular, there are 8 fixations and 7 saccades on the scanpath in Figure 2. In order to calculate the scanpath duration, their durations should be summed up. Scanpath dura-

tion is associated with scanning efficiency; therefore when the scanpath duration is longer, scanning becomes less efficient [24, 57, 20].

2.1.2 Frequency and Regression Metrics

Many frequency metrics exist. These metrics mainly depend on the number of fixations and saccades. The number of fixations overall is associated with the number of items which people need to process [24]. If people are asked to find only one item on the Web page and they make many fixations, it indicates less efficient search [24]. While searching, people make saccades. When they make more saccades, it means they search more [24].

If an AoI consists of more fixations, it shows that the AoI is more noticeable and significant compared to others [58]. Similarly, the number of gazes per AoI indicates the significance of the AoI [35].

On-target fixations are calculated by dividing the number of fixations on the particular area by the total number of all fixations [24]. The results can relate to the search efficiency where small values indicates less efficient search [24].

If people make a saccade and go back in the direction from which they have come, the second saccade can be defined as a regressive saccade and these saccades usually indicate the difficulty of extracting information from visual stimuli [63]. Regressive saccade is categorised under the regression metrics.

2.1.3 Spatial Metrics: Saccade and Scanpath Lengths

Longer saccades are more meaningful because the users' attention is drawn after longer distance [25]. Moreover, scanpath length, which is the sum of the lengths of all saccades on the scanpath, is associated with search efficiency [25]. When people make a larger scanpath, their searching are less efficient [25].

2.2 Eye Tracking Data Analysis with Visualisation Techniques

Heat maps and Gaze plots are two major techniques for visualising eye tracking data [54]. Although these techniques provide benefits for scanpath analysis, they have some limitations. Heat maps do not consider sequential patterns. In contrast, gaze plots take sequential patterns into consideration but when many gaze plots are shown on a Web page, it is difficult to analyse them because of the complex representation. These techniques are discussed below.

2.2.1 Heat maps

Heat maps consist of different colours where red usually represents the highest number of fixations or the longest time, in contrast, green represents the least [78]. Mostly fixated areas can be identified easily and these areas should be in common patterns. Figure 4 shows a heat map generated from a number of people's eye movements.

Heat maps have widely been used to analyse eye tracking data for different purposes such as investigating reading patterns [53, 31] and effects of representation of search results [17]. F-Shaped pattern was discovered as a dominant reading pattern [53]. This pattern indicates people rarely look other areas which are not in the boundaries of the F Shape [53].

Figure 4: Heat map generated from a number of people's eye movements on the Web page

In order to investigate this reading pattern, the eye movements of 232 participants were tracked while they were looking thousands of Web pages [53]. Figure 5 shows three Web pages as examples from the study.

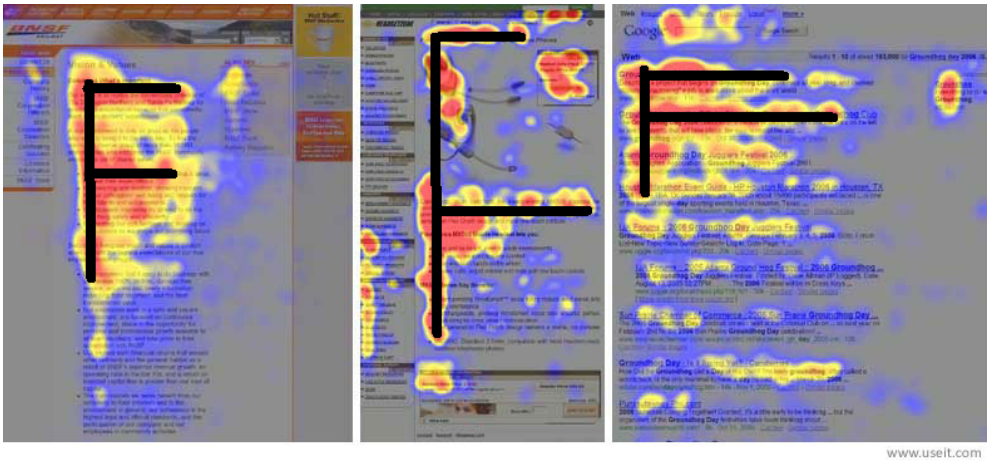


Figure 5: F-Shaped pattern as a reading pattern on Web pages [53]

Heat maps were also used to investigate how people read search engine result pages (SERPs) [31]. Golden Triangle pattern, shown in Figure 6, was discovered.

Some studies show that gender can affect heat maps on online dating sites [4]. For instance, [4]'s eye tracking study aimed to find areas which male and female users attend at most on online dating sites. Their study indicates that female viewers spend more time on profile information (such as demographics and interests) whereas male viewers spend more time on photos. These findings may be useful for designers working on online dating sites.

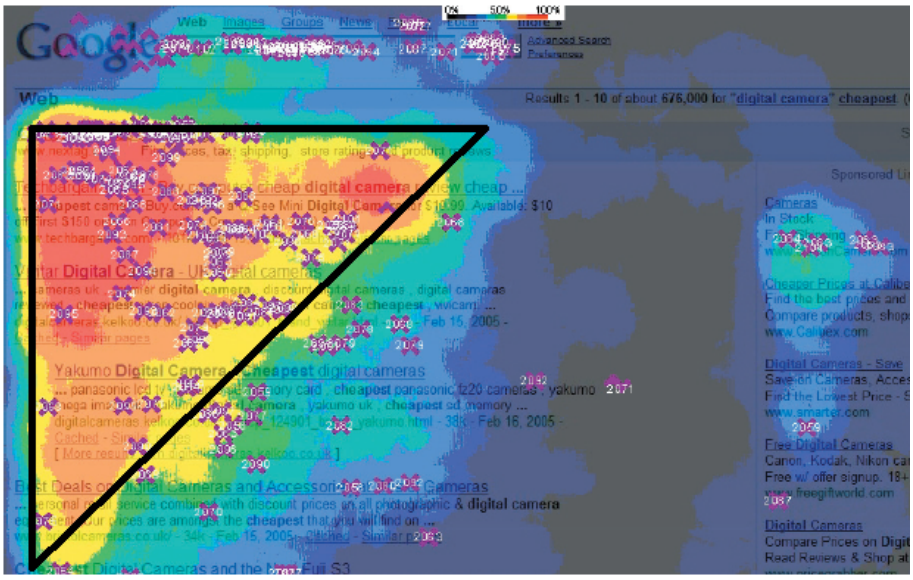


Figure 6: Golden Triangle as a reading pattern for search results on Google [31]

Figure 7 shows an example online dating Web site from this study for the male and female viewers.

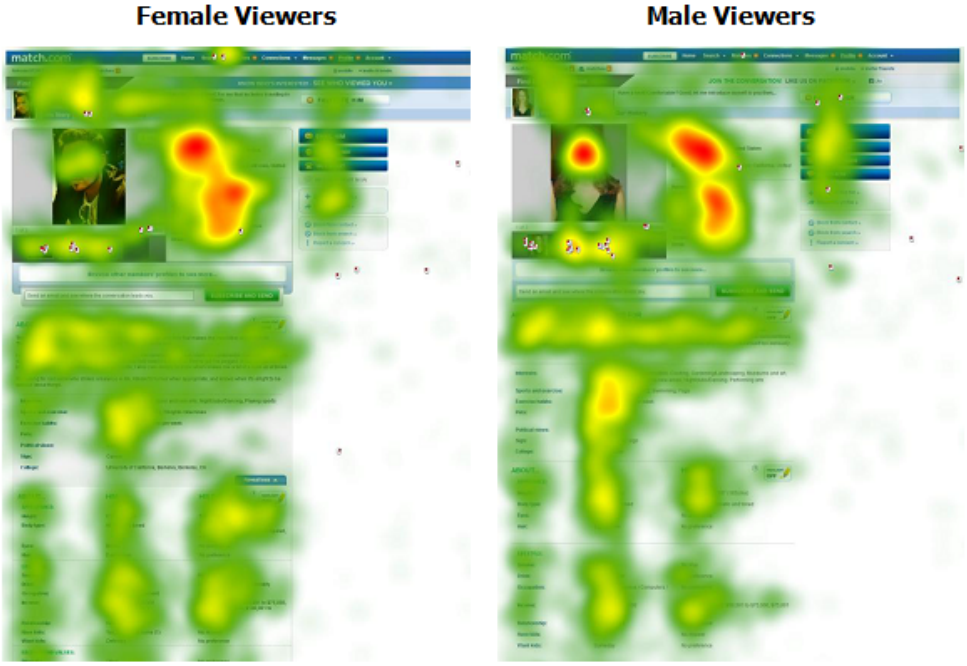


Figure 7: Heat maps of the male and female participants on the online dating Web site [4]

In order to investigate the impacts of the representation changes of search results, an eye tracking study was conducted by Cutrell and Guan (2007). The participants were required to follow some tasks which were categorised into two types: informational and navigational

Welcome to the Oklahoma City Zoo http://www.cpb.ouhsc.edu/OKC/OKCZoo/
Short
The oldest zoo in the Southwest and one of the top in the nation, the Oklahoma...
Medium
The oldest zoo in the Southwest and one of the top in the nation, the Oklahoma City Zoo's 110 acres are home to more than 2,800 of the world's most exotic animals.
Long
The oldest zoo in the Southwest and one of the top in the nation, the Oklahoma City Zoo's 110 acres are home to more than 2,800 of the world's most exotic animals." The Cat Forest/Lion Overlook was completed in 1997. New in 1993 was the Great EscApe , a simulated tropical forest with gorillas, orangutans and chimpanzees. Also found at the zoo are the Noble Aquatic Center: Aquaticus, a Children's Zoo and Discovery Area, Herpetarium, Island Life Exhibit, Dan Moran Aviary and the Safari Tram. Open 9-5 (Oct-March), 9-6 (April-Sept). Rides additional (weather permitting and seasonal). 2101 N.E. 50th Street Oklahoma City, OK (405) 424-3344 (OKC Zoo Phone Directory)

Table 2: Short, medium and long descriptions for search engine result pages [17]

tasks. Information tasks are related to finding specific information such as the address of the Aalborg Airport. On the other hand, navigational tasks are associated with finding a specific Web site such as the home page of Yahoo People Search. Both of these types of tasks need searching. The search results are shown to the participants with short (1-2 lines), medium (about 3-4 lines) and long descriptions (more than 4 lines). Example representation of the results are shown in Table 2. This study concludes that long description is useful for informative tasks. However, it causes some problems for navigational tasks. For instance, URLs are significant for navigational tasks but if the descriptions are long, URLs become far away from the title if they are located under the descriptions. It is recommended to locate URLs under the titles and above the descriptions to counteract with the problem.

Another eye tracking study use the same type of tasks and heat maps to examine how user behaviours vary when target results were displayed at various location [27]. This study shows that users rarely looked at lower ranking results, so they spent more time for searching to find a target which is located relatively low in the first page of search results, particularly for informative tasks. The participants highly agreed that they expect the information they are looking to be in the top five results. Besides, this study indicates that the participants re-queried without any click when they could not find a navigational task target. In contrast, the participants rarely re-queried without any click in the case of not finding an informative task target.

Heat map studies have been carried out for different purposes. These studies show that some areas of Web pages cannot catch people's attention [27, 31, 53]. Therefore, these areas should not be in common scanpaths. For example, the common scanpath ABCDE has been produced but the heat map shows that no one looked C. This situation indicates a problem in somewhere.

In brief, there are some metrics which have impacts on heat maps such as gender and task. For example, there is a considerable difference between female and male viewers' heat maps on online dating Web sites [4]. Moreover, the heat maps for informational and navigational tasks are not the same [17]. Therefore, these metrics should be considered

while analysing scanpaths because there might be differences in how people traverse on Web pages. Significant areas which caught people's attention can be recognised easily with heat maps but the sequence of these areas cannot be identified. This is a considerable limitation because we cannot understand where people firstly looked and which path they followed. Gaze plot becomes critical because it keeps sequences.

2.2.2 Gaze Plots

Gaze plot is a scanpath representation of one user [78]. Figure 2 illustrates an example gaze plot. It shows clearly which points are fixated and which path is followed. However, it is difficult to analyse multiple gaze plots using this visualisation technique as shown in Figure 8. Although the areas which did not catch attention can be recognised easily, it is impossible to analyse such representation in order to find common patterns in eye tracking data. Tobii Studio does not support any other visualisation techniques to show an aggregated scanpath for multiple users. Gaze plot actually represents a scanpath. Scanpaths have been analysed with a number of methods in the literature. These methods are explained in depth in the Section 2.3.

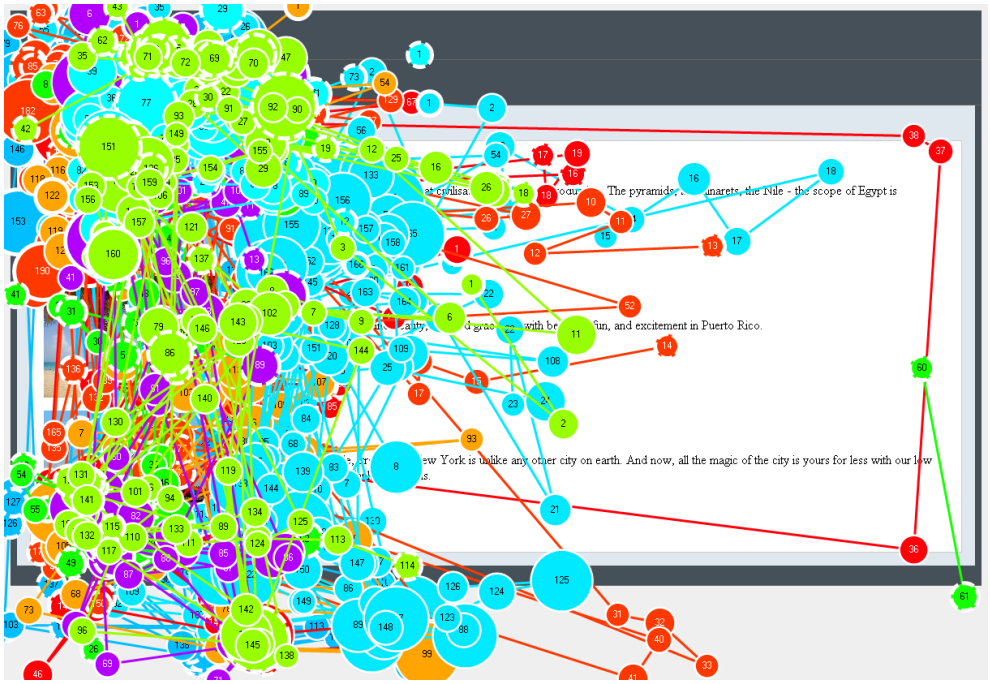


Figure 8: Multiple gaze plots on a Web Page

2.3 A Framework to Investigate Scanpath Analysis Methods

The existing approaches can be categorised into three main categories: analysing scanpaths as strings, geometric figures and series of salient elements.

In order to analyse, the existing methods used for scanpath analysis we have developed a framework as shown in Table 3. In this framework we have investigated whether or not these

methods support AoI inequality, multiple scanpaths, duration metrics, frequency metrics, regression metrics and spatial metrics. These metrics are explained in the Section 2.1.

Itti and Koch (2000) proposed using two-dimensional saliency maps to predict users' scanpaths. They pointed out that the sequence can be created in decreasing order of the saliency of objects and then this sequence can be considered as a scanpath. The saliencies of objects are determined based on intensity, colour and orientation. Example saliency map is shown in Figure 9. Saliency maps were tested with some existing scanpath analysis methods and the results showed that this approach did not always produce accurate results [21]. Since saliency maps were not successful, they will not be discussed in detail any more within this report.

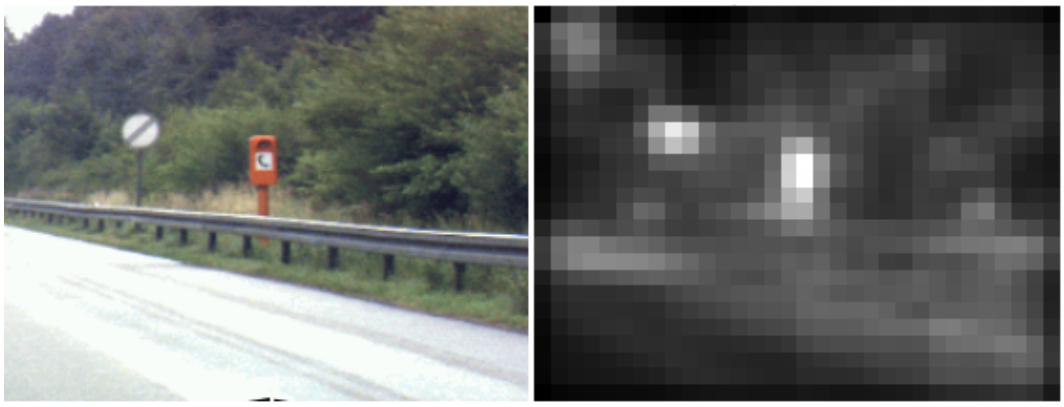


Figure 9: (a) Original image (b) Saliency map of the image [34]

2.3.1 Scanpath as Strings

The existing methods which analyse scanpaths as strings are categorised into five groups according to their goals: similarity calculation, transition between AoIs, super-pattern detections, sub-pattern detections and common scanpath identification.

Similarity Calculation: The Levenshtein Distance algorithm, which is widely known as the String-edit algorithm, was developed by Levenshtein (1966). Although the algoritm is quite old, many researchers still use and extend it to analyse scanpaths [38, 22, 19, 55]. The String-edit algorithm calculates the distance (i.e., dissimilarity) between two strings by finding the minimum number of operations (insertion, deletion and substitution) which are required to transform one string into another [75]. The dissimilarities can be used to categorise scanpaths [82]. It can also used to investigate differences between the behaviours of people on Web pages [38]. The pseudo-code of the algorithm is shown in Algorithm 1 where all operation costs are set to 1 (one) [80].

Assume that the String-edit algorithm tries to calculate the distance between ABCD and ABCE. This algorithm calculates the distance between these strings as 1 because the transformation can be done by only substituting D with E. To calculate the similarity between two strings the distance is divided by the length of the longer string to have a normalised score and then this score is subtracted from 1 [21]. Hence, the similarity between ABCD and ABCE is calculated as 75 % where the normalised score is 0,25 (1/4).

Scanpath Analysis Methods							
	Methods	AoI Inequality	Multiple Scanpaths	Duration Metrics	Frequency Metrics	Regression Metrics	Spatial Metrics
1. Scanpaths as Strings							
Similarity Calculation	Levenshtein Algorithm (String-edit Distance) [38, 43, 22, 19, 55]	✘	✘	✘	✘	✘	✘
	String-edit with Substitution Matrix [75]	✔	✘	✘	✘	✘	✘
	Vector String Editing [22]	✘	✘	✘	✘	✘	Saccade Len.
	Needleman and Wunsch Algorithm [52]	✔	✘	✘	✘	✘	✘
	ScanMatch [16]	✔	✘	Fixation Dur.	✘	✘	✘
	iComp [29]	✘	✘	✘	✘	✘	✘
Transition between AoIs	eyePatterns: Transition Matrix [82]	✘	✔	✘	# of Fix. per AoI	✔	✘
Super-patterns Detections	Shortest Common Supersequence [60]	✘	✔	✘	✘	✘	✘
Sub-patterns Detections	eyePatterns: Local Alignment Method [82]	✘	✘	✘	✘	✘	✘
	eyePatterns: Discover Patterns [82]	✘	✔	✘	✘	✘	✘
	eyePatterns: Search Patterns [82]	✘	✔	✘	✘	✘	✘
	eSeeTrack [79]	✘	✔	Fixation Dur.	# of Fix. per AoI	✔	✘
	T-Pattern [49, 45]	✘	✔	✘	✘	✘	✘
Common Scanpath	Averaging Scan Patterns [28]	✘	✔	✘	✘	✘	✘
	Dot-plots based Algorithm [23]	✘	✔	✘	✘	✘	✘
2. Scanpaths as Geometric Figures							
Similarity Calculation	Nearest Neighbour Method [6, 47]	✘	✘	✘	✘	✘	Saccade Len.
	Mathot et al. Method [50]	✘	✘	Fixation Dur.	✘	✘	Saccade Len.
	Vector-based, Multidimensional Scanpath Similarity Measure [36]	✘	✘	Fixation Dur.	✘	✘	Saccade Len.
3. Scanpaths as a Series of Salient Regions							
Scanpath Prediction	Saliency Map [21, 33]	✘	✘	✘	✘	✘	✘

Table 3: Framework to investigate scanpath analysis methods where ✘ represents "not supported" and ✔ represents "supported"

Algorithm 1 Levenshtein Distance Algorithm

Input: *char* string1[1..m] *char* string2[1..n]
Output: *int* dissimilarity

```
1:  int d[m, n]
2:  for i = 0 to m do
3:      d[i, 0] = i
4:  end for
5:  for j = 0 to n do
6:      d[0, j] = j
7:  end for
8:  for j = 1 to n do
9:      for i = 1 to m do
10:         if string1[i] = string2[j] then
11:             d[i, j] = d[i-1, j-1]
12:         else
13:             d[i, j] = minimum (d[i-1, j] + 1, d[i, j-1] + 1, d[i-1, j-1] + 1)
14:         end if
15:     end for
16: end for
17: return d[m,n]
```

Another example which illustrates how the String-edit algorithm works for "levenshtein" and "meilensthein" is shown in Figure 10. This example shows that there may be more than one way to transform one string into another with a minimum number of operations where "=" means match, "o" means substitution, "+" means insertion and "-" means deletion.

Scanpaths should be represented as strings to be analysed by the String-edit algorithm, so it is needed to divide the display or visual environment into AoIs [75]. While creating string representations of scanpaths, each fixation on a scanpath gets the name of its AoI [36]. The string representations of the scanpaths which are shown in Figure 11 are as follow [36]:

- Figure 11 (A): A6 C5 F0 J1 K2 I3
- Figure 11 (B): M M T C C H G M
  where M means Main, T means Title, C means Crawler, H means Headline and G means Globe.

This way of creating string representations of scanpaths allows keeping the sequence of fixations [36]. For example, the first fixation of the scanpath in Figure 11 (A) falls into A6 and the second fixation falls into C5. Thus, its string representation starts with A6 C5. It is important for us because we are interested in how people traverse on Web pages and which path they follow.

On the scanpath in Figure 11 (B), three fixations fall into the same AoI whose name is M. These fixations are not close to each other geometrically but they are related semantically [36]. If the scanpath runs on the grid-layout, the three fixations fall in the different AoIs. Hence, geometrically different scanpaths may have the same string representation

		m	e	ı	l	e	n	s	t	e	ı	n
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

l	e		v	e	n	s	h	t	e	i	n
o	=	+	o	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

or

l	e	v		e	n	s	h	t	e	i	n
o	=	o	+	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

Figure 10: How the String-Edit algorithm works for two sequences [1]

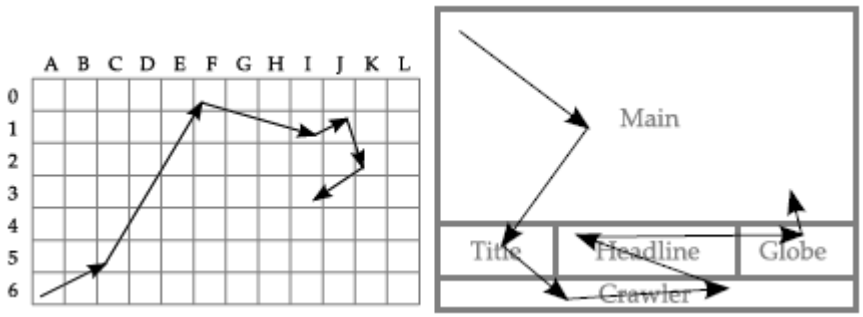


Figure 11: (A) Gridded AoIs and (B) Semantic AoIs [36]

when they run in semantic AoIs [36]. Grid-layout can be used to divide Web pages into AoIs which usually have the same size [36]. It has two main problems. Firstly, this layout may create AoI which does not have anything (just empty space). Secondly, inappropriate division can occur. Semantic division provides more meaningful AoIs. If researchers are interested in analysing scanpaths based on their meanings instead of geometrical properties, they should study with semantic AoIs.

Although the String-edit algorithm is able to find the minimum number of operations which are required to transform one string to another, it can only be useful for calculating dissimilarity or similarity between pairs of scanpaths. This algorithm is used as part of iComp which is a scanpath comparison tool [29]. iComp can accept more than two scanpaths and then determine the dissimilarities between those scanpaths based on pair-wise manner. These dissimilarities allows clustering scanpaths where similar scanpaths are located in the same cluster [29]. iComp uses fixation-based AoI identification to create string

representations of scanpaths [29]. This AoI identification technique uses fixation distribution [68]. If an area involves a number of fixations above a particular threshold, it can be defined as an AoI [68]. It was developed based on the mean shift procedure [68].

Inequality between AoIs, ignoring duration metrics and spatial information are considerable weaknesses of the String-edit algorithm. These weaknesses and the techniques to address them are explained below.

AoIs may have different sizes and the distances between AoIs are not the same. For instance, the distance between the header and the menu is different from the distance between the header and the footer in Figure 2. Therefore, the substitution cost between AoIs should not be the same because eyes make longer movements for longer distances. To address inequality between AoIs, Substitution matrix was proposed which can be created by using the following ways [75, 38]:

1. Setting a cost equal for all pairs of areas;

2. Setting a cost lower for areas which fall into related categories (such as, site navigation and content navigation) and setting a cost higher for areas which fall into different categories (such as, navigation types and content types);

3. Setting a cost lower for areas which are close to each other where the distance can be calculated by using different ways such as City Block and Euclidian Distance cost functions,

AoIs cannot be differentiated with the first way, so it is not useful to eliminate inequalities between AoIs. The adjacency approach can be used for the third way by setting the substitution cost lower for adjacent areas [38]. Assuming that the grid-layout in Figure 11 (A) will be used. The cost between A0 and L6 should be higher than the cost between E2 and F3 [38].

The third approach was used by Takeuchi and Habuchi (2007) to create a substitution matrix by using the City Block (see Equation (1)) and Euclidian Distance (see Equation (2)) cost functions. These functions determine substitution cost between areas u and v by using the following formulas where u_1 and u_2 are x and y coordinates of the centre of area u and α is a type of normalisation parameter [75].

$$f(u,v) = \alpha \sum_{i=1}^{2} |u_i - v_i| \tag{1}$$

$$f(u,v) = \alpha \sum_{i=1}^{2} \sqrt{(u_i - v_i)^2} \tag{2}$$

The Needleman and Wunsch algorithm was originally developed to find similarities in amino acid sequences of two proteins by aligning them globally [52]. Similar to the String-edit algorithm, it tries to transform one string to another by using the three types of operations which are addition, deletion and substitution [52]. However, this algorithm tries to maximise the alignment score, which can be considered as the similarity, between sequences instead of minimising the dissimilarity [69]. Substitution matrix is used so that

this algorithm is able to address inequalities between AoIs [16]. However, if this algorithm is used without a substitution matrix, it cannot differentiate AoIs [18].

Gap penalty can also be used with a substitution matrix [16]. It is usually a constant value which is used when aligning a character with a space [16]. Algorithm 2 shows the pseudo-code of the Needleman and Wunsch algorithm to calculate the alignment score where c is a gap penalty [40]. In addition, Figure 12 shows an example about how this algorithm works with using a substitution cost matrix and gap penalty for two sequences aAaB and aAaC. The alignment score may be different for identical scanpaths because of their lengths, for instance, longer identical scanpaths produce larger scores [16]. This problem can be solved using a normalised score [16]. Equation (3) shows the way of calculating a normalised score [16].

$$NormalisedScore = \frac{Score}{max(SubstitutionMatrix) * TheLongestSequenceLength} \quad (3)$$

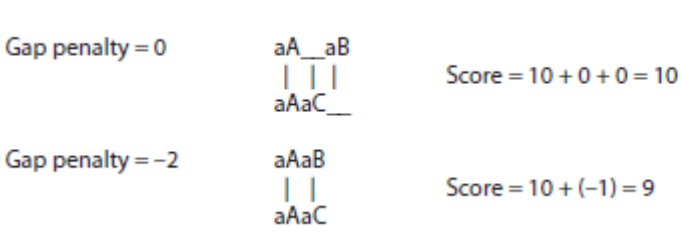
Algorithm 2 Needleman and Wunsch Algorithm

Input: *char* string1[1..m] *char* string2[1..n] *int* c *int* cost[1..m , 1..n]
Output: *int* similarity

```
1:  int d[m, n]
2:  for i = 0 to m do
3:      d[i, 0] = c * i
4:  end for
5:  for j = 0 to n do
6:      d[0, j] = c * j
7:  end for
8:  for j = 1 to n do
9:      for i = 1 to m do
10:         if string1[i] = string2[j] then
11:             d[i, j] = d[i-1, j-1]
12:         else
13:             d[i, j] = maximum (d[i-1, j] + c, d[i, j-1] + c, d[i-1, j-1] + cost(string1[i],
                string2[j]))
14:         end if
15:     end for
16: end for
17: return d[m,n]
```

ScanMatch is a scanpath comparison method which was developed based on the Needleman and Wunsch algorithm [16]. This method creates its own substitution matrix by using the Euclidian Distance cost function where the cost is maximum for the same AoIs and minimum for the AoIs that are the furthest from each other [16]. According to Cristino et al. (2010), when a substitution matrix is constructed well, gap penalty can be set to zero to prevent any inappropriate local alignments in order to provide the larger alignment score for global alignment.

Ignoring duration metrics is another significant weakness of the String-edit algorithm. As mentioned before, duration metrics can be interpreted in different ways. For example,



Substitution matrix

	aA	aB	aC
aA	10	−1	−5
aB	−1	10	−1
aC	−5	−1	10

Figure 12: How the Needleman and Wunsch algorithm works for the two sequences aAaB and aAaC with a substitution matrix when a gap penalty is equal to 0 and -2 [16]

long fixation duration can be interpreted as the difficulty for extracting information from visual stimuli [57]. Although these metrics are useful for scanpath analysis, the conventional String-edit and Needleman and Wunsch algorithms do not take these metrics into consideration. However, ScanMatch consider fixation durations. Temporal binning, which is particular time duration, is introduced and it causes the repetitions of the names of AoIs in string representations of scanpaths [16]. Figure 13 shows how the algorithm deals with fixation durations. 50 msec is defined as a temporal binning for this example. Since the duration of fixation A is 100 msec, the string representation of the scanpath starts with two As. Following this, 4 Cs and 5 Bs are added to the string representation. Thus, the scanpath is represented as AACCCCBBBBB. Note that remaining duration, which is less than 50 msec for this case, is ignored. For example, B has 260 msec, so 10 msec is remaining and it is ignored.

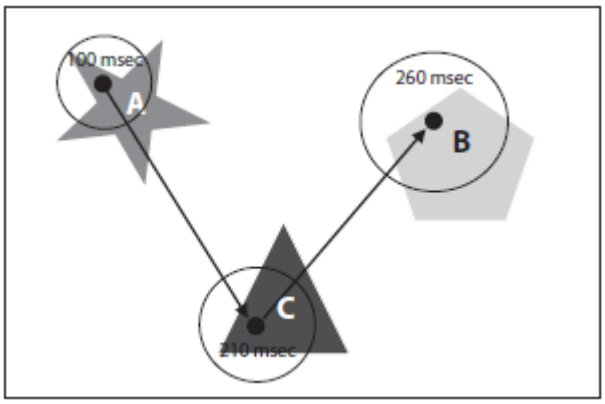


Figure 13: Creating the string representation of the scanpath by ScanMatch method with considering fixation duration [16]

Ignoring spatial information, which is related to saccades' lengths and angles, is another weakness of the String-edit algorithm. Spatial metrics can also be interpreted, for instance, when people follow a larger scanpath, their searching are less efficient [25]. The Vector String Editing method does not need any pre-defined AoIs because it creates string representations of scanpaths by using saccades instead of fixations [22]. There are n direction intervals for the saccade angles and m length intervals for the saccade length, so the angle and length of saccades on the scanpath are used to create string representations of scanpaths [22]. Since scanpaths are represented as strings, the conventional string-edit

algorithm can be applied to analyse them [22]. Therefore, it can be said that this algorithm tries to prevent ignoring spatial information by using saccades' lengths and angles.

Transition between AoIs: These methods and algorithms explained above have been applied to scanpaths in pair-wise manner but this project aims to find common patterns in a group of scanpaths. One of the techniques used is the Transition Matrix which is created using more than two scanpaths [82]. It calculates the transition probabilities between AoIs. The Transition Matrix was developed based on Markov Analysis which can be used for prediction [48]. Markov Analysis has been used in various areas such as finance, production and marketing [70]. In particular, it can be used to predict customers' buying behaviour for the marketing purposes [70].

An example transition matrix is shown in Figure 14 for four AoIs: A,B,C and D. This transition matrix was created for five sequences: ABACD, BACACAD, BDCACA, ABDADACACA and CDCDABD. As shown in the figure, Transition Matrix consists of table cells. Each cell has three rows where the first row shows the number of occurrences, the second row shows the row probabilities and the third row illustrates the column probabilities. Row probabilities allow identifying the next AoI and column probabilities allow identifying the previous AoI of the particular AoI. For example, according to the transition matrix, if the people look AoI D, it is most likely that they will look AoI A (60 %) after AoI D and they have already looked AoI B (60%) just before AoI D.

	A	B	C	D	Total
A	0	3	6	2	11
	0.0%	27.28%	54.55%	18.19%	100%
	0.0%C	100.0%C	75.0%C	25.0%C	
B	2	0	0	3	5
	40.0%	0.0%	0.0%	60.0%	100%
	18.19%C	0.0%C	0.0%C	37.5%C	
C	6	0	0	3	9
	66.67%	0.0%	0.0%	33.34%	100%
	54.55%C	0.0%C	0.0%C	37.5%C	
D	3	0	2	0	5
	60.0%	0.0%	40.0%	0.0%	100%
	27.28%C	0.0%C	25.0%C	0.0%C	
Total	11	3	8	8	30

Figure 14: Transition matrix generated from the sequences ABACD, BACACAD, BDCACA, ABDADACACA and CDCDABD [82]

The common scanpath might be constructed by using the highest probabilities in the transition matrix. For example, the common scanpath may be started with A. Since C has the highest row probability in A's row, C will come just after A. These are followed by A again. Finally, the common scanpath is constructed as ACACAC... The end point is not known. Furthermore, it is not certain that A is the start point. Therefore, some significant problems arise: What is the start and the end point of the common scanpath? and Which probabilities should be considered?

Super-pattern Detection: To address these problems mentioned above, some other methods have been proposed. The shortest common super-sequence (SCS) method has

been mentioned in the literature to detect super-patterns in eye tracking data for more than two people [60]. Assume that S_1 and S_2 are two scanpaths and T is their shortest common super-sequence. In this case, deleting zero or more characters from Z should be able to give S_1 and S_2 [60]. Figure 15 shows how this method works.

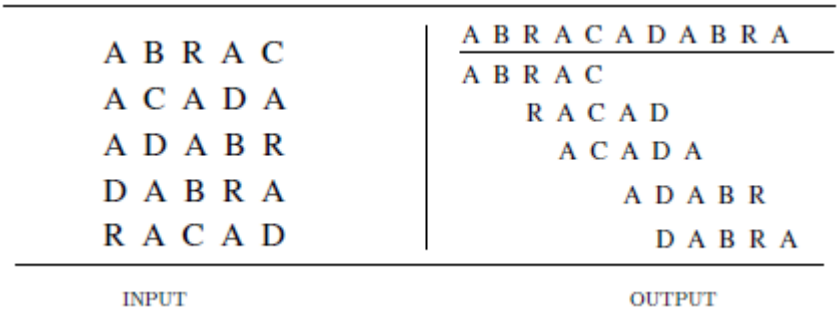


Figure 15: The shortest common super-sequence of the five sequences [71]

This method, however, does not always produce an efficient result [60]. For example, it produces the common scanpath ABA for the scanpaths AB and BA [60]. Thus, the common scanpath illustrates that the person looked one AoI firstly [60]. After that, s/he looked another AoI and then looked again the AoI which s/he firstly looked [60]. This situation does not exist for the both scanpaths AB and BA [60]. Furthermore, this technique does not always produce natural results [60]. For instance, there are four scanpaths whose string representations are ABCA, ABDA, ABEA and ABFA [60]. Although all of these scanpaths consist of four characters, this method produces a long scanpath which is ABCDEFA for them [60]. This example illustrates that the SCS method sometimes produces a result which does not comprise the significant features of the scanpaths (The length of the scanpaths will be completely different from the common scanpath for the given example) [60].

Sub-pattern Detection: Some methods try to detect sub-patterns in eye tracking scanpaths. The local alignment method is one of these methods [82]. It is also one of the methods of eyePatterns software application [2] [82]. The local alignment method was originally used in bioinformatics for finding patterns in data [72]. This method is capable of locally aligning two scanpaths using the Smith-Waterman algorithm [72]. This method is different from the Needleman and Wunsch algorithm because it focuses on local alignment instead of global alignment [72]. The local alignment method uses two scanpaths and highlights the sub-patterns which are seen in both of the scanpaths [82]. Only the areas which will give the highest similarity score are aligned [82]. A sample run is shown in Figure 16.

```
Local alignment of sequences
Score: 22.0          (22 matchs, 0 mismatches, and 31 gaps)
CA---CDCDC--AC--FDEBECACD--EBDEBEB-EB-EB-E--CEDEC-E-C
||   ||||   |   ||    | |  | |  |  |  |  |  | | | | |
CABFACDCD-EDA-EDFD----A-DFAE-D--E-DE-DE-DEDAC-D-CDEDC
```

Figure 16: A local alignment of two sequences [82]

One of the weaknesses of this method is dealing with only two sequences at the same

[2] http://sourceforge.net/projects/eyepatterns/

time [82]. Besides this, since this method tries to produce the largest similarity score, some sub-patterns cannot be recognised [82, 49]. Suppose that ABCDVWXYZ and VWXYZ-ABCD are two scanpaths. This method aligns VMXYZ because its similarity score is higher than the similarity score of aligning ABCD. This example is shown in Figure 17.

```
Local alignment of sequences
Score: 5.0          (5 matchs, 0 mismatches, and 0 gaps)
VWXYZ
|||||
VWXYZ
```

Figure 17: A local alignment of ABCDVWXYZ and VWXYZABCD [82]

eyePatterns also have a method to discover patterns [82]. It can deal with more than two scanpaths [82]. However, this method is quite restricted while discovering patterns [82]. For example, it cannot detect any sub-pattern for ACBE, ABE, ADBE. These scanpaths actually have ABE as a sub-pattern but this method does not have any tolerance for the additional letters such as (A (D) BE).

eyePatterns allows searching patterns in scanpaths and then returning a number of sub-patterns with their occurrences, too [82]. It is more tolerant in comparison to the discover patterns method by allowing gaps [82]. In particular, when ABE is searched with allowing gap size 1, the result will show that it occurs in all of three scanpaths: ACBE, ABE, ADBE. If the gap size was defined as 0 (zero), the result would show that ABE is seen only in the scanpath ABE.

eSeeTrack is an eye tracking analysis tool which considers three aspects: duration, frequency and sequence [79]. It visualises eye tracking data by using a timeline and hierarchical structure as shown in Figure 18 and Figure 19 [79]. A timeline shows the sequence of fixations where each fixation is represented as a coloured band [79]. The width of a band represents the duration [79]. Therefore, long fixations can be easily recognised in the timeline [79].

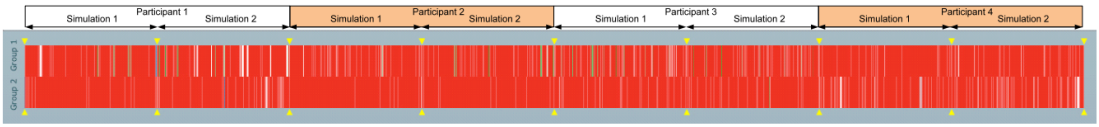


Figure 18: eSeeTrack - Timeline [79]

The hierarchical visualisation allows to see the sequential patterns with highlighting the probabilities [79]. For example, Figure 19 illustrates when people look the “Sales Promotion” area, they are likely to look at the same area again. [79].

T-Pattern, which stands for Temporal Pattern, was developed by Magnusson (2000) in the area of behavioural science for analyse social interaction. T-Pattern detection requires an input which is a series of events. According to Magnusson (2000), the type of behaviour is called as event type and the occurrence of the behaviour is called as event. In particular, “looks AoI A” is an event type and “looks AoI A at 10m:10s” is an event. Two event types are defined as a T-Pattern if the following two rules are satisfied [49]:

1. Both two event types occur at least twice in the behavior record in the same order.

Figure 19: eSeeTrack - Hierarchical visualisation which show the sequence with highlighting probabilities [79]

2. Their occurrences are invariantly distributed over time.

There are two possible types of distribution which is called as critical intervals (CI) [49]. First one is Fast Critical Interval which requires the event type A occurs just before the event type B [49]. Another critical interval is Free Critical Interval which requires the event type A occurs before the event type B within a defined time interval [49]. Thus, the event type A does not have to occur just before the event type B to be a T-Pattern for Free Critical Interval [49]. Figure 20 shows how critical intervals work.

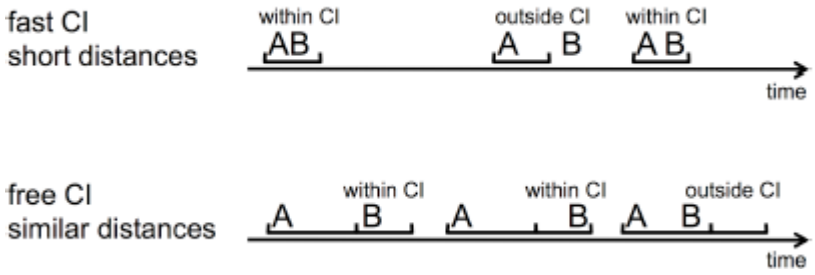


Figure 20: T-Pattern's Fast and Free Critical Intervals with examples [49]

As shown in Figure 21, each detected T-Pattern can be combined with another event type or T-Pattern [49]. As a consequence, a longer T-Pattern can be constructed by considering the significance level which affects how many event types can occur between T-Pattern elements while searching for critical intervals [49].

Figure 22 illustrates the longest T-Pattern which has been found for three users on Amazon Web site[3] [49]. They set minimum occurrences and length of T-Pattern to 2. Moreover, they preferred to use Fast Critical Intervals and set the significance level to 0.01. Since the critical interval and significance level configurations restrict detecting many T-Patterns, they may cause losing some sequential pattern information.

Common Scanpath Identification: Using multiple sequence alignment algorithm was proposed to identify a average scan pattern as a common scanpath for multiple users. According to this approach, each scanpath, which is a sequence, is aligned with another scanpath to have their alignment [28]. This process is repeated for all of the scanpaths until a

[3] `http://www.amazon.de/`

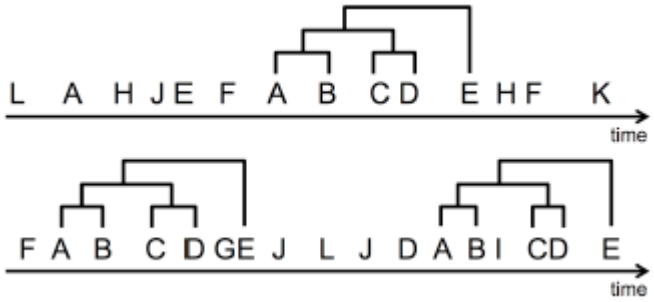


Figure 21: Longer T-Pattern detection with combining shorter T-Patterns [49]

Figure 22: Three users' common sub-pattern on Amazon home page detected by T-Pattern [49]

single scanpath is left and this scanpath is the average scan pattern which represents the common behaviour of the entire group of scanpaths [28]. However, this approach was not validated [28].

The Dot-plots algorithm was proposed for identifying common scanpaths for multiple users [23]. The Dot-plots algorithm was originally developed for comparing two biological sequences [42]. Figure 23 shows how the conventional Dot-plot algorithm works for the two sequences BDCEBCDCDCDEDEDCD and BCECDCDCDEDECDC. As shown in the figure, it uses two-dimensional matrix and one of the sequence is written horizontally whereas another one is written vertically. If the same characters are matched, a point was put into the intersection of these characters. After that, the longest line which represents the common sequence of two sequences is tried to be identified. In Figure 23, BCCDCDDED which is shown as a black line has been found as a common sequence of the two sequences. However, the red line can be longer than the black line but there are some disconnections

on the line. Because of this reason, it can be said that this algorithm is reductionist.

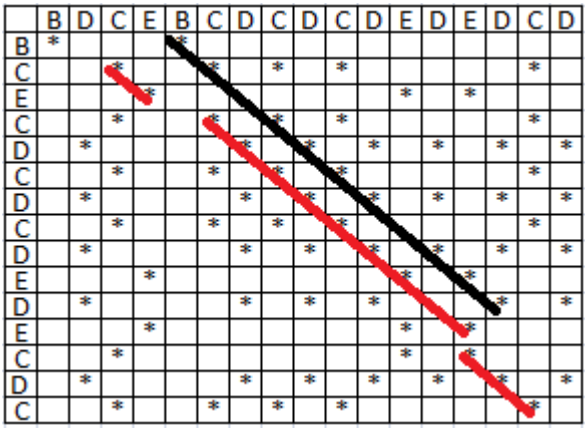


Figure 23: How the Dot-plot algorithm works for the two sequences BDCEBCDCDCDED-EDCD and BCECDCDCDEDECDC

The Dot-plots based algorithm proposed by Goldberg and Helfman (2010) uses a hierarchical structure where the scanpaths are leafs and the common scanpath is the root of the hierarchical structure. To create the hierarchical structure, two scanpaths which produce the longest common scanpaths with the Dot-plots algorithm are merged until a single scanpath is left. While finding common scanpaths, a pair of scanpaths is represented in the dot-plots and then find collinear points using regressions [23]. If the matching AoIs are close sufficiently to the regression line which is a sequence, these AoIs should appear in the same sequence [23]. Hence, some statistical models have been applied to address the reductionist approach of the Dot-plots algorithm.

2.3.2 Scanpath as Geometric Figures

The spatial information, especially the lengths and angles, are typically lost when scanpaths are analysed as strings. The following methods calculate similarities between scanpaths by taking this type of information into consideration. However, these methods do not tend to identify common patterns in eye tracking data. These methods are briefly explained below.

The nearest neighbour method was proposed by Mannan et al. (1995). This method is able to find the linear distance and then calculate the similarity between two scanpaths with preserving spatial information such as saccades lengths [47, 21]. The following equations (Equation (4) and (5)) are used to calculate the similarity of two scanpaths [47, 21].

$$D^2 = \frac{n_1 \sum_{j=1}^{n_2} d_{2j}^2 + n_2 \sum_{i=1}^{n_1} d_{1i}^2}{2n_1 n_2 (a^2 + b^2)} \tag{4}$$

$$Similarity = (1 - \frac{D}{D_T}) * 100 \tag{5}$$

where

- d_{1i} represents the distance between i^{th} fixation on the first scanpath and its nearest fixation on the second scanpath.

- d_{2j} represents the distance between j^{th} fixation on the second scanpath and its nearest fixation on the first scanpath.
- n_1 represents the number of fixations on the first scanpath.
- n_2 represent the number of fixations on the second scanpath.
- a and b represents the display dimensions.
- D_r is the mean linear distance which is calculated by using 100 randomly generated pairs of scanpaths.

The nearest neighbour method was improved by adding a constraint which states that each fixation on a scanpath can be a nearest neighbour of only one fixation on the second scanpath [30]. The nearest neighbour method and its improved version do not consider fixation durations [47, 30]. An additional improvement was made to support multi dimensions such as fixation durations by Mathot et al. (2012). This method is capable of supporting multi dimensions such as x and y coordinates, time and fixation durations [50]. Therefore, while trying to find distance between the fixations, these aspects of the fixations can be used [50]. The distance is calculated by the Euclidian distance shown in Equation (6) where n represents a number of dimensions is used [50]. For example, if fixation duration, x and y coordinates are used, there are three dimensions, so n is equal to 3.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{6}$$

When the distances are found, the distance between scanpaths S and T can be calculated by using the formula which is shown in Equation (6) [50]. n_s and n_t represents the length of S and T scanpaths [50]. In addition, $d^i{}_s$ is the distance between the fixation i in S to its closest fixation in T and $d^j{}_t$ is the distance between fixation j in T to its closest fixation in S.

$$D(S,T) = \frac{\sum_{i=1}^{n_S} d_s^i + \sum_{j=1}^{n_T} d_T^j}{max(n_S, n_T)} \tag{7}$$

The Vector-based, Multidimensional Scanpath Similarity Measure is able calculate the similarity between two scanpaths [36]. It does not use pre-defined AoIs to construct string representations of scanpaths because that way causes losing significant features of real scanpaths such as its shape, length, saccades' directions, fixations' positions and durations [36]. In order to simplify the representation of the scanpaths, saccade vectors with similar direction and/or short saccades vectors at similar positions are merged, then their fixation durations are added to the beginning of the newly created vector [36]. Figure 24 shows an example of original scanpath and its simplified version.

The saccades vectors on the simplified scanpath are matched the saccade vectors on another simplified scanpath as shown in Figure 25 [36]. After that, these matched saccade vectors can be compared with using the metrics such as saccades' shapes and lengths [36].

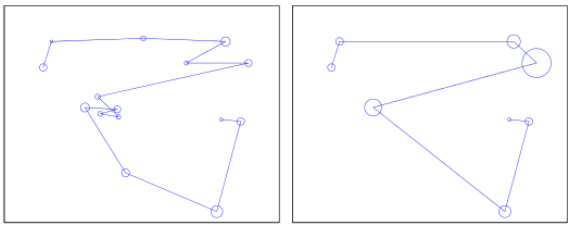


Figure 24: (A) Original scanpath (B) Simplified scanpath by the Vector-based, Multidimensional Scanpath Similarity Measure [36]

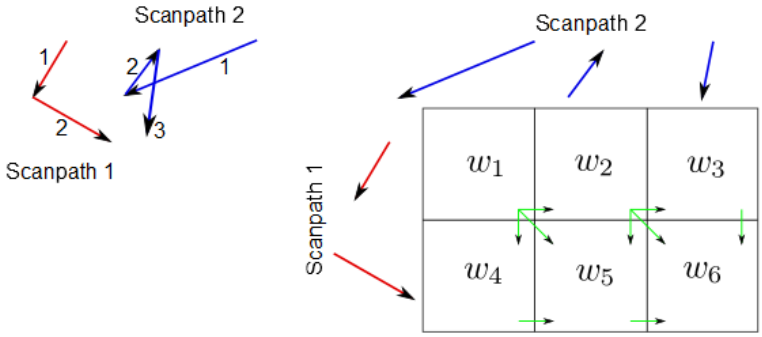


Figure 25: Matched saccades vectors of scanpaths by the Vector-based, Multidimensional Scanpath Similarity Measure [36]

2.3.3 The Weaknesses of the Existing Scanpath Analysis Methods

As can be seen in Table 3, none of the existing scanpath analysis methods support all of the metrics. Although there are some promising methods - The averaging scan patterns method [28] and the Dot-plots based algorithm [23] are two examples, these methods do not consider any eye tracking metrics mentioned in the Section 2.1. However, these metrics are useful to analyse data to increase its quality, for example, very short meaningless fixations can be excluded [64, 73]. In addition, these metrics can be used to detect and eliminate outliers which typically decrease the commonality in scanpaths.

Both of these methods suggest to use hierarchical structure. However, the averaging scan patterns method which proposes using multiple alignment was not validated [28]. In addition, it was not explained which sequences should be merged [28]. Because of the hierarchical structure, some information in intermediate levels can be lost because of merging two scanpaths [15]. Assume that there are three sequences: S1:GATACCAT S2:CTAAAGTC and S3: GCTATTGCG [15]. S1 and S2 can be aligned firstly and then S1'= - - A - A - - A - - - can obtained [15]. Following this, S1' and S3 can be aligned and then S3'= - - - A - - - - - - - - can be obtained [15]. This example illustrates that the hierarchical structure can make the method reductionist. Hence, this issue should be taken into consideration.

2.4 Possible Scanpath Analysis Methods

This project is interested in common sequential patterns in eye tracking data of multiple users on Web pages. In the literature, there is not much research that takes more than two scanpaths and tries to understand how people traverse a Web page for finding common patterns in scanpaths (see Table 3). The Dot-plots algorithm [23] and eyePatterns's discover patterns technique [82] are two examples but these methods produce shorter scanpaths. Short scanpaths are not useful for transcoding Web pages. In particular, when a common scanpath consists of only one area, how this common scanpath can be used to transcode Web pages?

There are some methods which have been successfully applied to find patterns in sequences in other fields, not in eye tracking [80, 15, 76]. These methods can potentially be applied to find common eye tracking scanpaths. This chapter explains how these methods work and how they provide benefits for scanpath analysis.

The longest common sub-sequence method can be used to produce longer sequences. This algorithm was originally developed as a dynamic programming method to find longest sub-sequence of two sequences, such as DNA Sequences, with time and space complexities O(n^2) [80, 15]. A sequence is called as a sub-sequence if it can be obtained by removing zero or more characters from the original sequence [15]. For instance, ABD is a sub-sequence of ABCD because it can be obtained by deleting one character, D, from the original sequence. AB and CD are also the sub-sequences of ABCD. The longest common sub-sequence (LCS) of two sequences is the longest sub-sequence among all common sub-sequences of both of the two sequences [15]. Since the LCS method tries to find the longest sequence shared by two sequences, it is considered as possible approach to address the reductionist problem of the existing methods.

Two steps are used in order to find the common sub-sequence of two sequences. The pseudo-code of these steps are illustrated in Algorithm 3 and 4 [4]. Besides, Figure 26 shows the longest common sub-sequence of five sequences and Figure 27 shows how the LCS method work for two sequences.

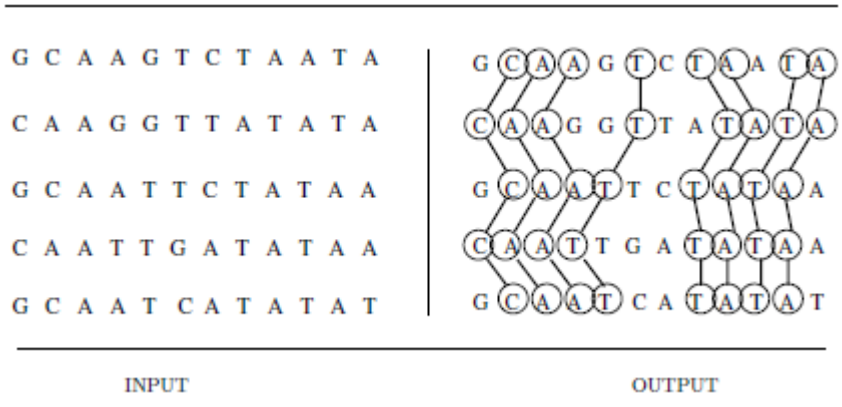


Figure 26: The longest common sub-sequence of the five sequences [71]

The pseudo-codes provided is able to find the LCS of two sequences which is called as 2-LCS [15]. Step 1 constructs a matrix and Step 2 use the matrix to trace back it to create a

[4] http://www.cs.umd.edu/~meesh/351/mount/lectures/lect25-longest-common-subseq.pdf

Algorithm 3 Longest Common Subsequence: Step 1

Input: *char* x[1..m] *char* y[1..n]
Output: *int* c[0..m, 0..n]

```
1:  int c[0..m, 0..n]
2:  for i = 0 to m do
3:      c[i,0] = 0
4:      b[i,0] = SKIPX
5:  end for
6:  for j = 0 to n do
7:      c[0,j] = 0
8:      b[0,j] = SKIPY
9:  end for
10: for j = 1 to m do
11:     for i = 1 to n do
12:         if x[i] == y[j] then
13:             c[i,j] = c[i-1,j-1]+1
14:             b[i,j] = ADDXY
15:         else if c[i − 1, j] >= c[i, j − 1] then
16:             c[i,j] = c[i-1,j]
17:             b[i,j] = SKIPX
18:         else
19:             c[i,j] = c[i,j-1]
20:             b[i,j] = SKIPY
21:         end if
22:     end for
23: end for
24: return c[m,n]
```

LCS of two sequences. 2-LCS can be solved by using these steps [15]. However, it is NP-hard (Non-deterministic Polynomial-time hard) to find k-LCS because k is not constant [15]. For instance, if the number of sequences is not constant, it is NP-hard to find the LCS of more than two sequences such as five sequences which are shown in Figure 26 [15]. As the number of sequences should be more than two and should not be constant, this issue should be considered.

The Long Run algorithm [7], the Expansion algorithm [37] and the Best Next for Maximal Available Symbols (BNMAS) [32] are three algorithms for k-LCS problem. These algorithms are explained with examples below [15]. Since this research project focuses on finding the common scanpaths of multiple users, these algorithms can be considered useful possible approaches.

The Long Run algorithm is explained with an example. Suppose that there are four sequences which are S1: GCCGAAGTGAGC, S2: AGCTACAGTGCT, S3: AGACATGTACGA and S4: ACGCAAGTGAGC. The number of occurrence of each symbol in each sequence is identified. Therefore, S1: [2A 5G 3C 2T], S2: [3A 3G 3C 3T], S3: [5A 3G 2C 2T] and S4: [4A 4G 3C 1T] are identified for the four sequences as a result of which the minimum number of occurrences are selected so that [A G C T]: [2A 3G 2C 1T] is

Algorithm 4 Longest Common Subsequence: Step 2

Input: *char* x[1..m] *char* y[1..n] *int* b[0..m,0..n]
Output: *String* LCS

```
1:  String LCS
2:  j = m
3:  j = n
4:  while i != 0 AND j != 0 do
5:    if b[i, j] = ADDXY then
6:      add x[i] to front of LCS
7:      i - -
8:      j - -
9:    else if b[i, j] = SKIPX then
10:     i - -
11:   else if b[i, j] = SKIPY then
12:     j - -
13:   end if
14: end while
15: return LCS
```

	-	A	T	C	G	A	G	G	T
-	0	0	0	0	0	0	0	0	0
T	0	0	1	1	1	1	1	1	1
C	0	0	1	2	2	2	2	2	2
A	0	1	1	2	2	3	3	3	3
G	0	1	1	2	3	3	4	4	4
T	0	1	2	2	3	3	4	4	5
T	0	1	2	2	3	3	4	4	5
C	0	1	2	3	3	3	4	4	5
G	0	1	2	3	4	4	4	5	5

	-	A	T	C	G	A	G	G	T
-	0	0	0	0	0	0	0	0	0
T	0	0	1	1	1	1	1	1	1
C	0	0	1	2	2	2	2	2	2
A	0	1	1	2	2	3	3	3	3
G	0	1	1	2	3	3	4	4	4
T	0	1	2	2	3	3	4	4	5
T	0	1	2	2	3	3	4	4	5
C	0	1	2	3	3	3	4	4	5
G	0	1	2	3	4	4	4	5	5

Figure 27: How the longest common sub-sequence method works: (a) Step 1 (b) Step 2 [15]

identified. This helps to understand all of the sequences have two As, three Gs, two Cs and one T. This algorithm does not provide anything about the sequence of the symbols which is essential for this project. However, this algorithm helps to identify mostly visited AoIs which can be interpreted as important and noticeable areas [58]. These areas should be seen in common scanpaths.

The Expansion algorithm is also explained with an example. Suppose that there are two sequences S1: AAAACCCAAAA CCA and S2: AAACCCCAAAACCC. These sequences are re-written, so S1 becomes $A^4C^3A^4C^2A$ and S2 becomes $A^3C^4A^4$ C^3. Therefore, ACAC is a sub stream which has no contiguously same symbol and the sub-stream is seen in both of the sequences. This sub-stream is expanded step by step, for example, A^2CAC is the first expanded sub-stream, and $A^3C^3A^4C^2$ has been found.

This technique provides a solution for k-LCS problem. Suppose that there are four

sequences which are S1: AABBCABCA, S2: BBCAABCBA, S3: CACACBBCB and S3: ABCACBACB. The longest common sub-sequences are found in pair-wise manner by using the technique which is mentioned just above. ACACBCB and BBCABCA have been found as LCSs. One of them should be chosen. Assume that ACACBCB is chosen which is the LCS of S3 and S4. In this case, S3 and S4 are removed from the list and their LCS should be put in the list. Hence, the list involves three sequences which are (S3, S4): ACACBCB, S1: AABBCABCA and S2: BBCAABCBA. This process is repeated until a single sequence is left. The single sequence is the common sub-sequence in all of the sequences. Particularly, some AoIs can be lost when the LCS of two sequences is tried to be found. Therefore, the lost AoIs cannot be evaluated in next stage. This can cause producing shorter sequences at the end.

Figure 28 shows how the Best Next for Maximal Available Symbols algorithm works. In this example, there are four sequences and it is tried to find their LCS. A, G, C and T are four different symbols which are used in these sequences. v(A), v(G), v(C) and v(T) should be calculated. Let us calculate v(T) as an example. Here, it is required to find v(T,1), v(T,2), v(T,3) and v(T,4). For v(T,1), the first sequence which is GCCGAAGTGGCT should be analysed. Last T should be selected and then the number of occurrences before this symbol (including the symbol) should be calculated. There are two As, three Cs, five Gs and two Ts. Hence, v(T,1) = [2 3 5 2] is calculated. By using the same way, v(T,2) is calculated as [3 3 3 3], v(T,3) is calculated as [3 2 1 2] and v(T,4) is calculated as [3 2 2 1]. After that, the minimum number of occurrences for each symbol is selected and the v(T) is calculated as [2 2 1 1]. v(A): [2 1 2 0], v(G): [2 3 2 1] and v(C): [2 2 2 1] are calculated, too. Since the highest sum of all values in v(Symbol) is v(G)'s, G is put into the LCS. Then, last G is found all of the sequences and the symbols which appear after the symbol (including the symbol) are removed and the process is continued until one of the sequence is lost. This algorithm considers frequency which is significant while analysing eye tracking data as mentioned earlier.

Our literature review shows that people can follow different paths while traversing on Web pages. Thus, the probabilities between AoIs should be considered. Apriori algorithm is another useful approach for finding patterns using a probabilistic approach. Market basket transactions are well known examples of these types of algorithms [76]. Table 4 illustrates an example dataset and the rule which can be extracted from the data set is $\{\text{Diapers} \rightarrow \text{Cola}\}$.

TID	**Items**
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Table 4: Example market basket transactions dataset [76]

The example below illustrates how Apriori algorithm works step by step with the dataset in Table 8.

1. Suppose that a database is read and then the frequent 1-itemsets and their supports

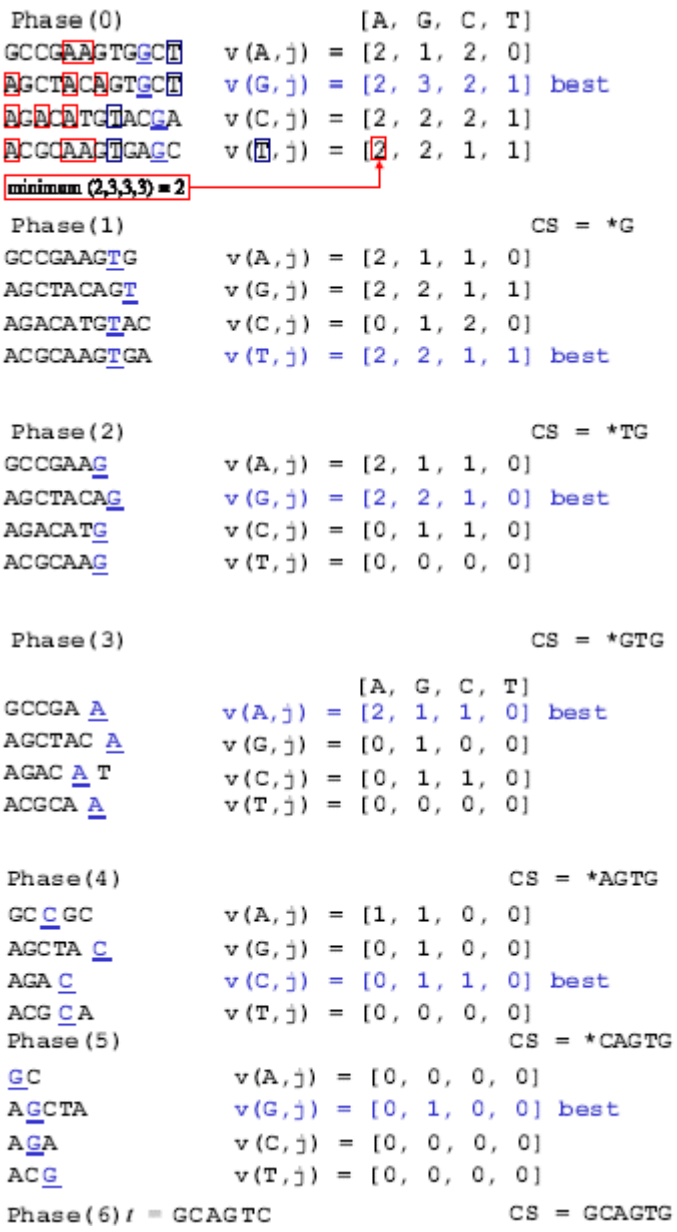


Figure 28: How BNMAS works [15]

are identified as shown in Table 8. Minimum support value is very significant here. Assume that it is defined as '2'. Hence, {1} and {9} are eliminated because of the minimum support value and then the following list is created. L1 = {{2} → 6, {3} → 6, {4} → 4, {5} → 8, {6} → 5, {7} → 7, {8} → 4}

2. Candidate generation step is applied to L1 and the following candidates are created. C2 = { {2,3}, {2,4}, {2,5}, {2,6}, {2,7}, {2,8}, {3,4}, {3,5}, {3,6}, {3,7}, {3,8}, {4,5}, {4,6}, {4,7}, {4,8}, {5,6}, {5,7}, {5,8}, {6,7}, {6,8}, {7,8}} The Prunning step, which will be explained later, does not affect the list C2. The database is read

{1}	2
{2}	6
{3}	6
{4}	4
{5}	8
{6}	5
{7}	7
{8}	4
{9}	2

Table 5: Example dataset (1-itemsets) for the demonstration of how Apriori algorithm works

again to identify the support of each candidate in C2. The following list is created because of the minimum support value. L2 = { {2,3} → 3, {2,4} → 3, {3,5} → 3, {3,7} → 3, {5,6} → 3, {5,7} → 5, {6,7} → 3}

3. Candidate generation step is applied to L2 and the following candidates are created.

   {2,3,4} is generated from {2,3} and {2,4}
   {3,5,7} is generated from {3,5} and {3,7}
   {5,6,7} is generated from {5,6} and {5,7}
   C3 = {{2,3,4}, {3,5,7}, {5,6,7}}

   {2,3,4} is pruned by the Prunning step owing to the fact that not all of the sub-sets of size 2 which are {2,3}, {2,4} and {3,4} do not exist in L2. Therefore,
   C3 becomes {{3,5,7}, {5,6,7}}

   The database is read again to identify the support of each candidate in C3. The following list created because of the minimum support value. L3 = {{3,5,7} → 3}

4. As L3 has only one element and C4 has no element, the algorithm stops and returns the following list L = L1 ∪ L2 ∪ L3

Apriori algorithm does not take sequential patterns into consideration [59]. It means that {3,5} and {5,3} are not different [59]. However, this project is trying to find common sequential patterns in eye tracking data for transcoding Web pages. PrefixSpan [56] (Prefix-projected Sequential pattern mining) and BIDE (BI-Directional Extension based frequent closed sequence mining) [81][5] are two algorithms which are similar to Apriori algorithm but consider sequential patterns with a probabilistic approach. However, BIDE discovers only closed sequential patterns in sequence databases.

2.5 Lessons Learned

To achieve our project's objective we need to analyse eye tracking data from different aspects in depth to find common patterns. This section briefly explains how eye tracking data

[5] http://www.philippe-fournier-viger.com/spmf/index.php

should be analysed step by step in order to achieve our objective.

Since users may differently traverse on Web pages [17, 27], it can be difficult to identify common sequential patterns in eye tracking data for multiple users. Eye tracking metrics will guide us to identify outliers. Assuming that eye tracking data shows us most people, such as 90 %, complete a specific task very quickly but only a few of them make many saccades to complete the task. The users should be analysed separately from the others to increase commonality. Besides, the users who cannot complete the task should be separated if any.

Eye trackers collect a large volume of data, however it is suggested that people need at least 150 ms to process object [64]. Therefore, the fixations whose durations are less than 150 msec may not be meaningful. Short fixations can be excluded to eliminate noisy data [73]. Pre-processing of eye tracking data helps improving the quality of eye tracking data.

This project aims to identify common scanpaths in terms of visual elements of Web pages. Web page's visual elements, i.e., AoIs, can be generated automatically. The segmentation algorithm should provide semantic AoIs instead of the simple grid-layout AoIs to have more meaningful AoIs.

When AoIs are identified, it can easily be identified which AoIs fixations locate using fixations' x and y coordinates and AoI's widths, lengths, x and y coordinates. Next, fixations get the name of their AoI to create string representations of scanpaths.

There is not much research that identifies common sequential patterns in eye tracking data for multiple users on Web pages (see Table 3). The existing ones, such as eyePatterns's discover and search patterns techniques [82], can produce very short scanpaths which cannot be useful for transcoding Web pages. The longest common sub-sequence method can be used to produce longer scanpaths [15]. In addition, the probabilities of transition between AoIs should be taken into consideration all of the users may follow different paths but a considerable number of people may follow the same path.

The existing scanpath analysis methods tend to ignore the complexities of underlying cognitive processes: when one follows a path to achieve a task, there is a reasoning that affects their decision, and none of these algorithms capture that. Cognitive processes should be taken into consideration.

3 Preliminary Experiments

Preliminary experiments were carried out to investigate the strengths and weaknesses of the existing scanpath analysis methods. These experiments will inform the development of our algorithm. Our algorithm will mainly be developed on the preliminary experiment results and the literature review findings, in particular, it will be based on our framework explained in the previous chapter. The existing eye tracking dataset was used [10]. This chapter starts with describing the dataset and then continues with the discussion of the conducted experiments.

3.1 An Eye Tracking Dataset

The existing eye tracking dataset collected by Brown at al. (2012) was used for our preliminary experiments. For the purpose of their study, Brown at al. (2012) designed a Web

site for HCW Travel Company. Figure 29 shows the home page of the HCW Travel Web site. They asked their participants to complete two tasks on the home page: (1) read latest news from HCW Travel Company and (2) click on "Special Offers" link on the main menu. While the participants were completing the tasks, their eye movements were tracked using Tobii Eye Tracker with the screen resolution 1280 x 1024. Twelve people participated who were students and employees at the University of Manchester between the ages of 18 and 45.

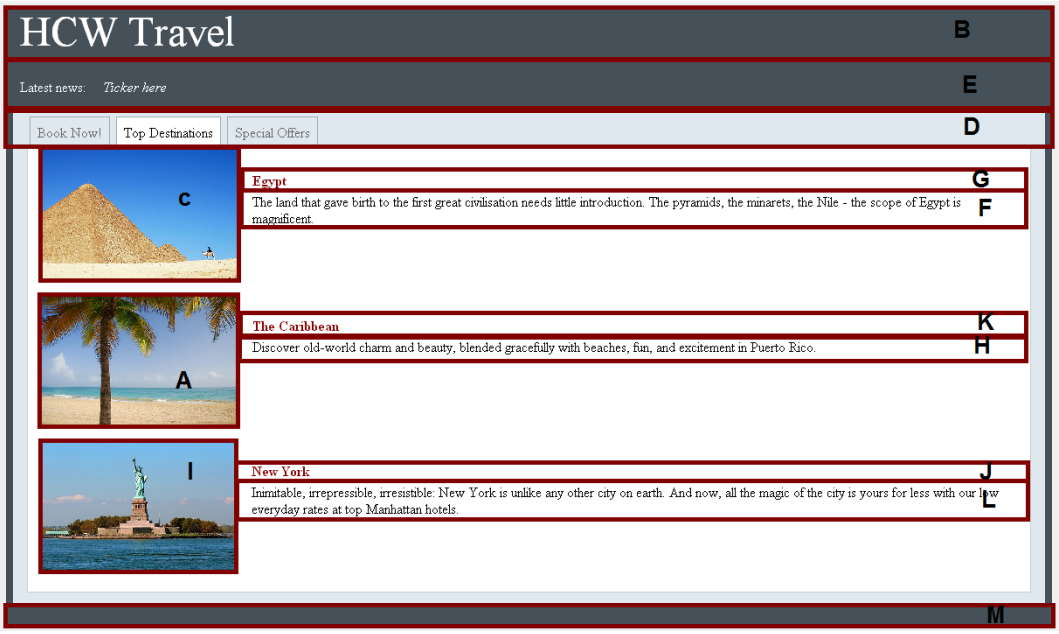


Figure 29: The HCW Travel Home Page

When we watched the eye tracking recordings, we recognised that there were some problems for two participants. One participant could not complete the tasks accurately because of some misunderstandings and another participant did not look the screen appropriately. Therefore, we eliminated their data from our experiments.

We needed AoIs to create the string representations of the scanpaths. These AoIs should be associated with the source code of Web pages because they will be used to transcode Web pages. Since VIPS is a well-known Web page segmentation algorithm which uses the structural information provided in HTML DOM and visual presentation [13, 83], we have created the AoIs automatically using the extended version of VIPS algorithm developed as a part of eMine project [6] [2]. The extended version supports some additional terms such as HTML5 Tags [2].

We developed an application on ACTF Platform [7] to create the string representations of the scanpaths using automatically generated AoIs. This application accepts eye tracking data in CSV format which includes fixation information such as fixation duration, x and y coordinates. Since we have already known AoIs' widths, height, x and y coordinates from the VIPS algorithm, our application easily detects which AoIs the fixations are located using

[6] http://emine.ncc.metu.edu.tr/
[7] http://www.eclipse.org/actf/

the x and y coordinates of fixations. Next, the fixations get the names of their AoIs to create the string representations. Our application is also able to highlight an AoI for a fixation, namely point, of a particular recording as shown in Figure 30.

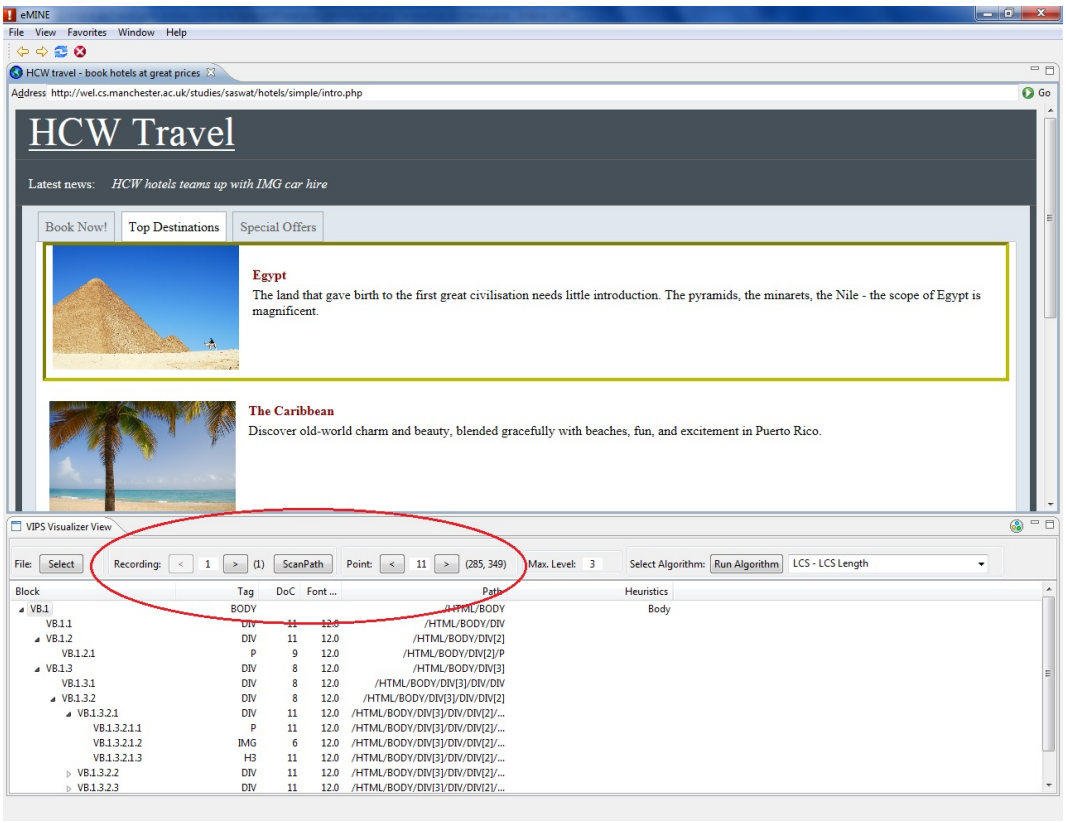


Figure 30: Our application for creating string representations of scanpaths

Once AoIs were automatically generated, the existing eye tracking dataset was exported on the home page of HCW Travel Company in CSV format. It was used as an input for our application. Since the 5th segmentation granularity level was found as the most successful level with approximately 75% user satisfaction, we decided to use the 5th level for our experiments [3]. To prevent any problems in identifying the fixations' AoIs, we studied with a screen whose resolution is the same as the Tobii Eye Tracker's screen resolution. Besides this, the string representations of the scanpaths were simplified by abstracting repetitions [9, 36]. For example, AABBBCCCCC becomes ABC. Table 6 shows the abstracted string representations of the scanpaths our application creates for the segmented Web page shown in Figure 29 without excluding any fixations.

As can be seen from the table, people have different strategies to complete the same task, and they follow slightly different paths to achieve the same task. For example, participant 3 and 4 followed a longer path in terms of the number of visual elements visited to complete the given task compared to participant 6 and 8.

Scanpaths	
S1:	ABCBDCACACBEBDEBE
S2:	FGFGHAEBEBDEBEBEDEDCECEDECDG
S3:	ABIJKGDFGFGCDEDAEDLJDADIEDEDEDEDEDAKGDCDEDGC
S4:	CFCFCGDGFCDFACIDEBECKCDEBDEBEBEBEBECEDECFECFD
S5:	DACFGFADEDEDEDEDEDC
S6:	DFDCABEDCD
S7:	BDGEBCGDFDCDEDEDCD
S8:	CFCGFCFCDECEFCD
S9:	BGEFGCDCDGDEDEFDC
S10:	DKHDGFBEFDGB

Table 6: The abstracted string representations of the scanpaths on the segmented Web page shown in Figure 29

3.2 Experiments with the Existing Approaches

This section discusses the experiments which were conducted with some existing scanpath analysis methods.

3.2.1 String-edit Algorithm

The String-edit algorithm can be used to determine the distance, i.e., dissimilarity, between two eye tracking scanpaths [43]. As it works with only two scanpaths at the same time, it can be applied to multiple scanpaths in pair-wise manner. Two-dimensional matrix can be used to store the dissimilarities between all pairs of the scanpaths.

We applied the String-edit algorithm to the scanpaths in Table 6. Figure 31 shows the result when the String-edit algorithm was applied to Scanpath 1 and 8 as an example [8]. In addition, Table 7 shows two-dimensinonal matrix which includes the dissimilarities between all pairs of the scanpaths.

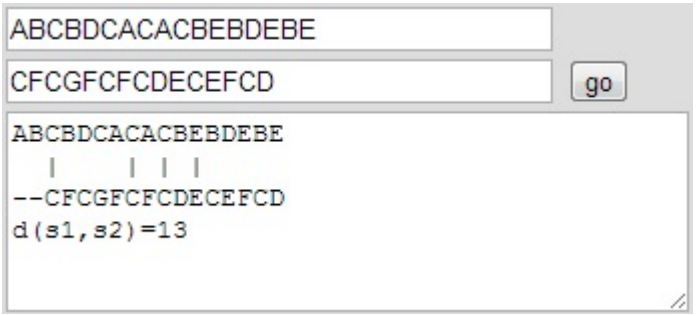


Figure 31: How the String-edit algorithm works for the two sequences

To conclude, since this project is interested in common patterns in eye tracking data, we may choose to create a hierarchical structure where the root shows the common scanpath for all of the scanpaths. In this case, similar scanpaths should be merged to produce longer

[8] http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	0.0	21.0	34.0	33.0	14.0	11.0	14.0	13.0	13.0	12.0
S2	21.0	0.0	27.0	25.0	16.0	21.0	17.0	21.0	19.0	23.0
S3	34.0	27.0	0.0	31.0	28.0	36.0	31.0	35.0	31.0	37.0
S4	33.0	25.0	31.0	0.0	30.0	36.0	33.0	30.0	33.0	38.0
S5	14.0	16.0	28.0	30.0	0.0	13.0	11.0	12.0	9.0	14.0
S6	11.0	21.0	36.0	36.0	13.0	0.0	10.0	10.0	11.0	7.0
S7	14.0	17.0	31.0	33.0	11.0	10.0	0.0	10.0	8.0	13.0
S8	13.0	21.0	35.0	30.0	12.0	10.0	10.0	0.0	11.0	13.0
S9	13.0	19.0	31.0	33.0	9.0	11.0	8.0	11.0	0.0	13.0
S10	12.0	23.0	37.0	38.0	14.0	7.0	13.0	13.0	13.0	0.0

Table 7: The distances between the scanpaths in Table 6 calculated by the String-edit algorithm

scanpaths to next level. Otherwise, unacceptably short or no common scanpath is likely to be produced. Using the two-dimensional matrix allows us identifying the two most similar scanpaths. For example, Table 7 illustrates that Scanpath 6 and 10 are the most similar scanpaths because the dissimilarity between two scanpaths is the lowest value (which is 7.0) in the matrix.

3.2.2 Transition Matrix

The Transition matrix method determines the probabilities of the transition between AoIs [82]. Since there might be differences in how people traverse on Web pages, it is useful to keep probabilities, for example, 90 % of the participants might AoI A after AoI B [17].

When this method was applied to the scanpaths shown in Table 6, the transition matrix in Figure 32 was created. For example, Figure 32 illustrates that when people look AoI D, they looked AoI E after AoI D with 46,16 % and looked AoI E before AoI D with 39,63 % probability. Although this matrix is able to show the transition probabilities between AoIs, it does not give any information about the start and the end points.

In conclusion, using a probabilistic approach is useful because of the possible variations in scanpaths. However, additional analysis should be done to calculate the probabilities about the start and the end points. For example, when we look the scanpaths shown in Table 6, we see that more people started with AoI D; therefore AoI D might be considered as a start point.

3.2.3 eyePatterns

We conducted experiments with the following three methods of eyePatterns application: the local alignment method, the discover patterns technique and the search patterns technique.

Local Alignment Method

The local alignment method focuses on locally aligning two sequences to maximise the alignment score, which can be considered as the similarity score [72]. This technique of

	A	B	C	D	E	F	G	H	I	J	K	L
A	0 0.0% 0.0%	3 25.0% 15.0%	4 33.34% 11.77%	2 16.67% 3.78%	2 16.67% 4.35%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 8.34% 25.0%	0 0.0% 0.0%
B	0 0.0% 0.0%	0 0.0% 0.0%	2 9.53% 5.89%	5 23.81% 9.44%	12 57.15% 26.09%	0 0.0% 0.0%	1 4.77% 5.27%	0 0.0% 0.0%	1 4.77% 33.34%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
C	3 9.1% 30.0%	2 6.07% 10.0%	0 0.0% 0.0%	12 36.37% 22.65%	4 12.13% 8.7%	7 21.22% 33.34%	3 9.1% 15.79%	0 0.0% 0.0%	1 3.04% 33.34%	0 0.0% 0.0%	1 3.04% 25.0%	0 0.0% 0.0%
D	4 7.7% 40.0%	0 0.0% 0.0%	10 19.24% 29.42%	0 0.0% 0.0%	24 46.16% 52.18%	4 7.7% 19.05%	7 13.47% 36.85%	0 0.0% 0.0%	1 1.93% 33.34%	0 0.0% 0.0%	1 1.93% 25.0%	1 1.93% 100.0%
E	0 0.0% 0.0%	13 28.89% 65.0%	7 15.56% 20.59%	21 46.67% 39.63%	0 0.0% 0.0%	4 8.89% 19.05%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
F	2 9.1% 20.0%	1 4.55% 5.0%	7 31.82% 20.59%	5 22.73% 9.44%	1 4.55% 2.18%	0 0.0% 0.0%	6 27.28% 31.58%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
G	0 0.0% 0.0%	1 5.56% 5.0%	3 16.67% 8.83%	5 27.78% 9.44%	2 11.12% 4.35%	6 33.34% 28.58%	0 0.0% 0.0%	1 5.56% 50.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
H	1 50.0% 10.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 50.0% 1.89%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
I	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 33.34% 1.89%	1 33.34% 2.18%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 33.34% 50.0%	0 0.0% 0.0%	0 0.0% 0.0%
J	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 50.0% 1.89%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 50.0% 25.0%	0 0.0% 0.0%
K	0 0.0% 0.0%	0 0.0% 0.0%	1 25.0% 2.95%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	2 50.0% 10.53%	1 25.0% 50.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%
L	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	0 0.0% 0.0%	1 100.0% 50.0%	0 0.0% 0.0%	0 0.0% 0.0%

Figure 32: Transition matrix generated by the scanpaths in Table 6

eyePatterns uses the following default values to determine the alignment score: match reward is 1, gap cost is 0 and mismatch penalty is -1 [82]. Moreover, it allows importing a substitution matrix for determining the alignment score. This method can be applied to two sequences. An example local alignment of our scanpaths 5 and 8 is shown in Figure 33.

The findings show that this technique detects some common patterns between two sequences but it does not support more than two sequences. The hierarchical structure can be constructed using this method for further experiments to see whether it is capable of producing useful results for our project.

```
Local alignment of sequences S8 and S5
Score: 8.0          (8 matchs, 0 mismatches, and 15 gaps)
CFCGFCFC-DEC-EF-------C        (0...13)
||  |   |  ||  |        |
CF-G--F-ADE-DE-DEDEDEDC        (2...18)
```

Figure 33: A local alignment of the scanpaths 5 and 8 in Table 6

Discover Patterns

eyePatterns application allows discovering patterns in multiple eye tracking scanpaths but it has no tolerance for extra items [82]. It asks for defining the length of pattern and minimum number of occurrences.

Figure 34 shows an example result when the length of pattern is set to 5 and the minimum number of occurrences is set to 2. According to Figure 34, DEDED is a pattern which is seen in eight times but in the three scanpaths which are our third, fifth and seventh scanpaths in Table 6. However, DEDED pattern cannot be detected in the ninth scanpath (BGEFGCDCDG**DEDEFD**C) because of the extra item F. This example clearly illustrates that this algorithm is very reductionist which is not efficient for our project.

```
Search within sequences S1, S2, S3, S4, S5, S6, S7, S8, S9, S10
Pattern             Seen      Sequences:
DEDED               8         3/10 30.0%
EBDEB               3         3/10 30.0%
EDEDE               6         2/10 20.0%
EBEBE               4         2/10 20.0%
DCDED               2         2/10 20.0%
CEDEC               2         2/10 20.0%
ECEDE               2         2/10 20.0%
EDEDC               2         2/10 20.0%
DEBEB               2         2/10 20.0%
BDEBE               2         2/10 20.0%
BEBDE               2         2/10 20.0%
```

Figure 34: The discover patterns technique results when it was applied on the scanpaths in Table 6 with the length of pattern 5 and the minimum number of occurrences 2

Search Patterns

eyePatterns application provides an opportunity to search patterns in eye tracking scanpaths, too [82]. This technique is more tolerable compared to the discover patterns technique because it has a gap size which allows maximum 5 extra items in within patterns. When DEDED was searched as a pattern without any gap size, it was found in three scanpaths as shown in Figure 35. In contrast, when the gap size is defined as 5, it was found in four scanpaths as shown in Figure 36.

To conclude, although this technique is useful for searching patterns with a gap size, our project does not focus on searching for specific patterns. However, when we find a common

```
The pattern was found in 3 sequences:
S3(1):
          DEDED (25...29)
S5(2):
          DEDED (7...11)
          DEDED (13...17)
S7(1):
          DEDED (11...15)
```

Figure 35: The search patterns technique results when it was applied on the scanpaths in Table 6 without any gap size

```
The pattern was found in 4 sequences:
S3(3):
          DEDAEDL (12...18)
          DIEDEDE (22...28)
          DEDEDA (29...34)
S5(2):
          DEDEDE (7...12)
          DEDEDC (13...18)
S7(1):
          DEDEDC (11...16)
S9(1):
          DEDEFDC (10...16)
```

Figure 36: The search patterns technique results when it was applied on the scanpaths in Table 6 with a gap size 5

scanpath for a set of scanpaths, we can use this technique to search for the common scanpath to see if it is supported by the set of scanpaths.

3.2.4 Dot-plots Algorithm

The conventional Dot-plots algorithm can be used to find the common sequence of two sequences [23]. It was applied to the following two scanpaths: BDGEBCGDFDCDEDEDCD and BGEFGCDCDGDEDEFDC. Figure 37 illustrates how this algorithm works for these scanpaths. Black line is detected as a common sequence. Although the red line can create a longer common sequence, it cannot be detected because of the disconnections within the line. This example shows that this method is not capable of detecting the possible longest scanpath.

As mentioned in the literature review, some statistical models were applied by Goldberg and Helfman (2010) to address the reductionist approach of the Dot-plots algorithm by merging the points into the line if they are close enough to the line. Moreover, Goldberg and Helfman (2010) points out that a hierarchical structure can be created using the Dot-plots algorithm with the statistical models to find a common scanpath for a set of scanpaths.

We constructed a hierarchical structure with the conventional Dot-plots algorithm and used the scanpaths in Table 6. D was found as a common scanpath which was not useful

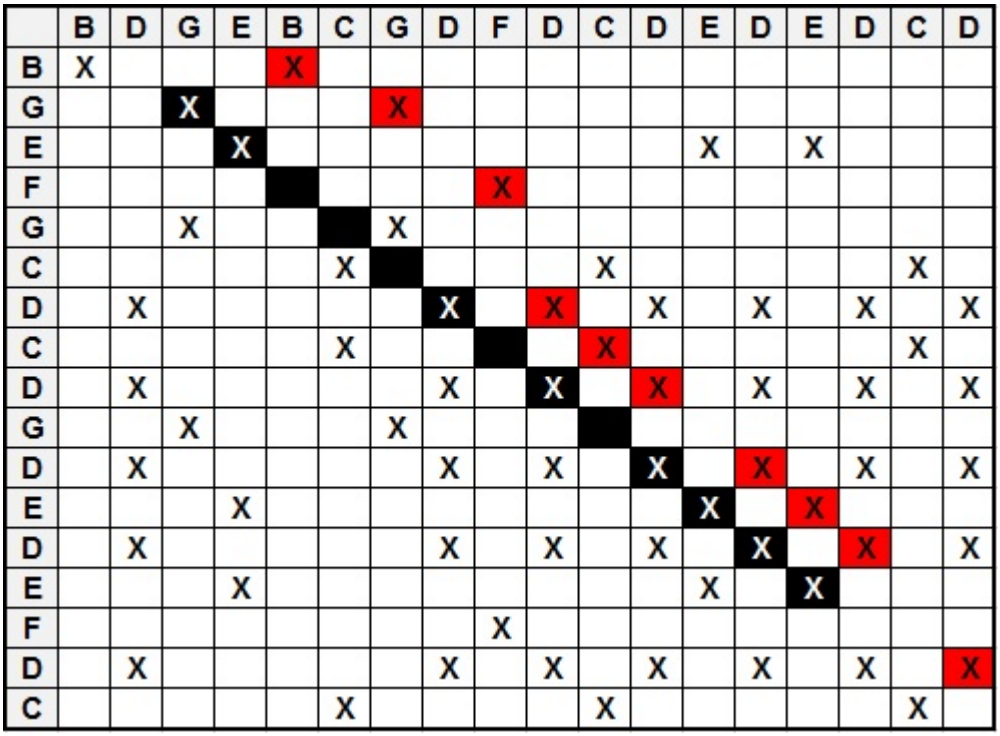


Figure 37: How the Dot-plots algorithm works for BDGEBCGDFDCDEDEDCD and BGEFGCDCDGDEDEFDC

for transcoding. Since the implementation of the method of Goldberg and Helfman (2010) is not available to the public, we require implementing it for further experiments.

3.3 Experiments with Possible Approaches

This section discusses the experiments conducted using possible scanpath analysis methods which were originally developed in other fields but can potentially be applied to eye tracking data (see the Section 2.4).

3.3.1 Longest Common Subsequence

The longest common sub-sequence method works with two sequences to find the longest sequence from all of the sub-sequences of two sequences [15]. When this algorithm was applied to the following two scanpaths: BDGEBCGDFDCDEDEDCD and BGEFGCDCDGDEDEFDC, it produces BGEFDCDEDEDC whereas the Dot-plots algorithm produces GEDDDEDE and the discover patterns technique of eyePatterns can find DEDE as the longest common scanpath. These examples show that the LCS method is less reductionist than the Dot-plots algorithm and the discover patterns technique of eyePatterns. Hence, it can be used to address the reductionist approach of other algorithms. However, the LCS method does not consider any probabilities, so it tends to ignore AoIs if they are not visited by all users, even though most users may visit them.

3.3.2 Apriori Algorithm

The particular implementation of Apriori algorithm was used for our preliminary experiments [8]. This algorithm was applied to the eye tracking scanpaths shown in Table 6 with the minimum support 75 %. The results in Table 8 were provided. These findings show that there is no repetition in the patterns, so each AoI can be seen once in the patterns. Therefore, it does not support regressive saccades which can be interpreted as the difficulty of extracting information from visual stimuli [63]. Furthermore, it can be seen that all of the participants looked AoI D and AoI E. Similarly, heat maps show which areas get more attention of users. However, both heat maps and Apriori algorithm do not give any information about their sequence.

As a result, we cannot find sequential patterns with Apriori algorithm. However, the probabilistic approach guides us which AoIs should be seen in common scanpaths. In particular, D and E should be seen in the common scanpaths. In addition, C should be considered because 90 % of the participants visited it.

Pattern	Support %
D E	100
D	100
E	100
B D E	80
B D	80
B E	80
B	80
G D E F	80
G D E	80
G D F	80
G D	80
G E F	80
G E	80
G F	80
G	80
C D E	90
C D	90
C E	90
C	90
C F D E	80
C F D	80
C F E	80
C F	80
F D E	90
F D	90
F E	90
F	90

Table 8: Apriori algorithm results when it was applied on the scanpaths in Table 6 with a minimum support 75%

3.3.3 BIDE Algorithm

BIDE is an algorithm which is developed based on Apriori algorithm, but it discovers closed sequential patterns in sequence databases [81]. The experiment was conducted with the minimum support 75 %. The findings are shown in Table 9. As can be seen from the table, 80 % of the participants looked AoI D and then AoI E, even though they may look some other AoIs between D and E. If D is far enough from E in the sequence, DE is not considered as closed sequential pattern. Although some information can be lost, closed sequential patterns can be useful for transcoding. For example, if there is a closed pattern DE, AoI D and AoI E should be close to each other in the transcoded version of the Web page.

Pattern	Support %
D E	80
D	100
E	100
F E	70
F	90
G E	80
B	80
C	90

Table 9: BIDE algorithm results when it was applied on the scanpaths in Table 6 with a minimum support 75%

4 eMINE Scanpath Analysis Algorithm

Our current work aims to address the problems of the existing scanpath analysis methods which are mentioned in the literature review (especially in the Section 2.3) and the discussion of the preliminary experiment results. Some common problems are being reductionist, ignoring eye tracking metrics and cognitive process. However, in the scope of eMINE project we focused on the problem being reductionist. We developed eMINE scanpath analysis algorithm that takes a number of scanpaths and returns a pattern that is common in all scanpaths that means we are trying to identify a route in terms of visual elements followed by our participants.

Algorithm 5 shows our proposed eMINE algorithm which takes a set of scanpaths and return a scanpath which is common in all the given scanpaths. If there is only one scanpath, it returns that one as the common scanpath, if there is more than one, then it tries to find the most similar two scanpaths in the given list. It does this by using the the Levenshtein Distance which is the traditional String-edit algorithm [38]. Then it removes these two scanpaths from the given list of scanpaths and introduces their common scanpath to the list of scanpaths given originally. This continues until there is only one scanpath.

4.1 eMINE Scanpath Analysis Algorithm with an Example

This process is illustrated in Figure 38 using the scanpaths in Table 10. These scanpaths were created using the 3rd segmentation granularity level without excluding any fixations. When we start the algorithm there are 10 scanpaths, then in the first iteration the most common scanpaths are S2 and S4, therefore we apply the longest common sub-sequence method to these two scanpaths and generate a common one called S24. S2 and S4 are then removed from the list and S24 is introduced to the list of scanpaths. For all ten scanpaths in Table 10, our algorithm returns CDED as the common scanpath which is illustrated in Figure 39 by arrows. It shows that people start to look at the first item of the content, then they looked at the menu, the latest news part and then they looked at the menu.

Algorithm 5 Find Common Scanpath

Input: Scanpath List
Output: Scanpath
1: **if** the size of Scanpath List is equal to 1 **then**
2: **return** the scanpath in Scanpath List
3: **end if**
4: **while** the size of Scanpath List is not equal to 1 **do**
5: Find the two most similar scanpaths in Scanpath List with the Levenshtein Distance
6: Find the common scanpath by using Longest Common Subsequence
7: Remove the similar scanpaths from the Scanpath List
8: Add the common scanpath to the Scanpath List
9: **end while**
10: **return** the scanpath in Scanpath List

	Scanpath
S1:	ABCBDCACACBEBDEBE
S2:	CAEBEBDEBEBEDEDCECEDECACDC
S3:	CABFACDCDEDAEDFDADFAEDEDEDEDEDACDCDEDC
S4:	CACDCDCACFDEBECACDEBDEBEBEBEBECEDECECD
S5:	DACADEDEDEDEDEDC
S6:	DCDCABEDCD
S7:	BDCEBCDCDCDEDEDCD
S8:	CACDECECD
S9:	BCECDCDCDEDECDC
S10:	CACDADCBECDCB

Table 10: The abstracted string representations of the scanpaths on the segmented Web page shown in Figure 39

4.2 Initial Informative Validation

If we revisit the task that the participants were asked to complete in the original study (see the Section 3.1), we see that the path generated by our algorithm is quite logical [10].

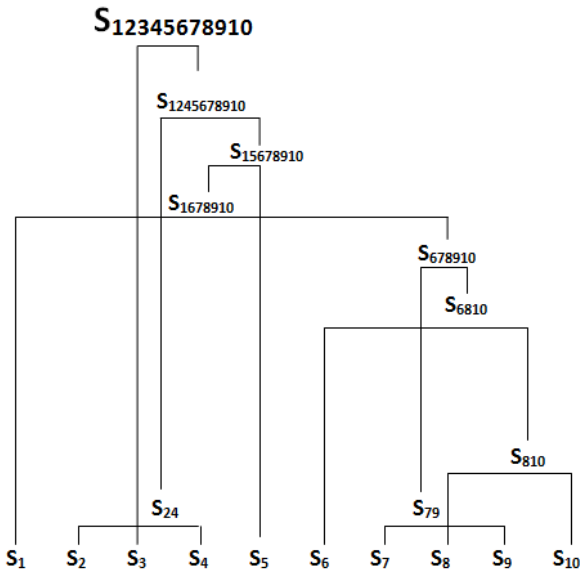


Figure 38: Our algorithm applied on scanpaths in Table 6.

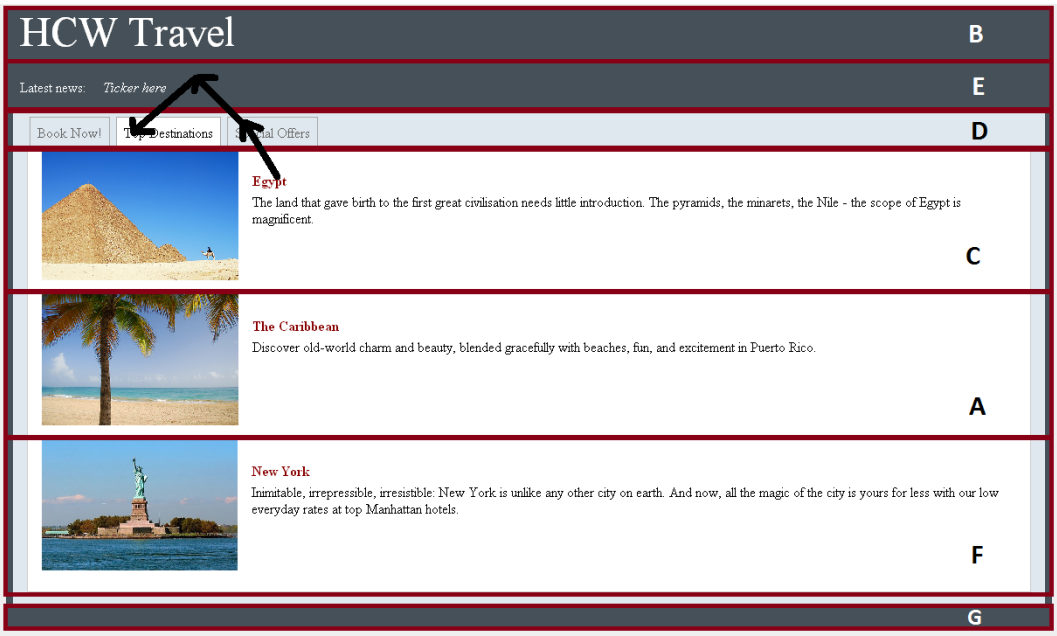


Figure 39: Common scanpath on the HCW Travel Web page detected by our algorithm

The path very nicely matches with what people asked to complete. When we generate a transition matrix, we also see that it supports this path. For example, according to the transition matrix shown in 11, if the people look AoI D, it is most likely that they will look AoI E (46.16 %) after AoI D and they have already looked AoI C (42.6 %) just before AoI D.

	A	B	C	D	E	F
A	0 0.0% 0.0%	3 15.0% 15.0%	11 55.01% 23.92%	3 15.0% 5.56%	3 15.0% 6.53%	0 0.0% 0.0%
B	0 0.0% 0.0%	0 0.0% 0.0%	3 14.29% 6.53%	5 23.81% 9.26%	12 57.15% 26.09%	1 4.77% 25.0%
C	12 25.54% 63.16%	4 8.52% 20.0%	0 0.00% 0.00%	23 48.94% 42.6%	7 14.9% 15.22%	1 2.13% 25.0%
D	5 9.62% 26.32%	0 0.0% 0.0%	21 40.39% 45.66%	0 0.0% 0.0%	24 46.16% 42.18%	2 3.85% 50.0%
E	0 0.0% 0.0%	13 28.89% 65.0%	11 24.45% 23.92%	21 46.67% 38.89%	0 0.0% 0.0%	0 0.0% 0.0%
F	2 50.0% 10.53%	0 0.0% 0.0%	0 0.0% 0.0%	2 50.0% 3.71%	0 0.0% 0.0%	0 0.0% 0.0%

Table 11: Transition matrix generated by the scanpaths in Table 10

When our algorithm is applied to the scanpaths in Table 6 which were created for the 5th segmentation granularity level, DED is created as a common scanpath. This common scanpath is also supported by the transition matrix in Figure 32 which is created using the scanpaths. It also supports the tasks on the Web page. As can be seen from the result, the segmentation granularity level is important. It affects the common scanpath. Therefore, the best suitable segmentation level should be identified for our project. Moreover, further studies with more participants and pages need to be conducted to validate the proposed approach.

5 Summary

To address the problems of accessing the Web in such constrained environments, transcoding techniques have been proposed to re-engineer Web pages to make them more accessible [83, 46, 39, 44]. If we know how people read Web pages, we can provide them the more efficient version of Web pages. Eye tracking may allow us to drive Web page transcoding with a better understanding of users' experience and prediction of future interactions. In order to make the Web pages more accessible for a wide range of people, we are interested in transcoding based on common patterns in eye tracking data instead of individual patterns.

However, there is not much research in identifying common scanpaths. The conventional Dot-plots algorithm [23] and eyePatterns's discover patterns technique [82] are two examples. However, these algorithms are reductionist which means they are likely to produce unacceptable short scanpaths for transcoding. Moreover, they tend to ignore cognitive process which can affect people's decisions. Besides, they simply accept string representations of scanpaths without analysing eye tracking data. Since eye trackers collect a large amount of data, pre-processing should be applied by considering eye tracking metrics to improve the quality of data. For example very short fixations, which have no meaning, can be removed [64, 73]. However, most of the algorithms do not focus on eye tracking metrics as shown in our framework (see Table 3). In the scope of eMINE project, we developed an algorithm to address the problem of being reductionist as mentioned in Section 4. The remaining problems will be considered as future work.

Glossary

Area of Interest
: Area of Interest which is widely known as AoI is an area on a display or visual environment which gets fixations [35]. The identification of these areas depends on researchers' purposes. For example, some researchers prefer to use a grid-layout [75] and some other researchers use some semantic AoIs [17]. Header and menu are examples of semantic AoIs., 2

Fixation
: Fixation is a position where eyes are relatively stable within some threshold of diameter over some minimum duration and with a velocity below some threshold [35, 65]. Velocity is the distance between the current gaze point and the previous or next gaze point [67]. These parameters are used to aggregate gaze points into fixations [77, 67]. Fixations are represented as circles. New information is usually obtained from fixations [62]. However, an object can sometimes be identified without any fixations because of the visual field which is split into foveal, parafoveal and peripheral regions where foveal region is the narrowest and peripheral region is the widest [62]., 1

Gaze
: Gaze is defined as "a series of consecutive fixations within an area of interest" [35, 65]., 7

Gaze Point
: Gaze point is a raw eye movement data point and it can be distinguished from other data points using the combination of its timestamp, x and y coordinates [77]., 1

Saccade
: Saccade is a quick eye movement between fixations [57]. Saccades are shown as lines between circles. Eyes cannot be really constant, i.e. they make small movements, because of constant tremor of the eyes. It is called as Nystagmus [62]. These are different from saccades., 1

Scanpath
: Scanpath is a defined as "a sequence of fixations and saccades" [57]. Since a scanpath shows a sequence, it allows to see where the scanpath starts, which path it follows and where it finishes., 1

References

[1] The Levenshtein-Algorithm. http://www.levenshtein.net/.

[2] Elgin Akpinar and Yeliz Yesilada. Vision Based Page Segmentation: Extended and Improved Algorithm. Technical report, Middle East Technical University Northern Cyprus Campus, 2012.

[3] Elgin Akpinar and Yeliz Yesilada. Vision Based Page Segmentation Algorithm: Extended and Perceived Success. In *the 1st international workshop on Engineering Mobile Web Applications (EMotions'2013) - in conjunction with the 13th International Conference on Web Engineering*, EMotions 2013, 2013.

[4] AnswerLab and Tobii Technology. Field-based Eye-Tracking Study Of Online Dating Sites. Technical report, AnswerLab, San Francisco, CA, USA, February 2012.

[5] Chieko Asakawa and Hironobu Takagi. Transcoding. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, A Foundation for Research, Human Computer Interaction Series, pages 231–260. Springer, London, 2008.

[6] Marina Bloj, Glen Harding, and Alan Chalmers. Exploring Eye Movements for Tone Mapped Images. In *Proceedings of SPIE-IS and T Electronic Imaging*, pages 72400O–1 – 72400O–11, San Jose, CA, USA, 2009.

[7] Paola Bonizzoni, Gianluca Della Vedova, and Giancarlo Mauri. Experimenting an Approximation Algorithm for the LCS. *Discrete Applied Mathematics*, 110:200–1, 1998.

[8] Christian Borgelt. Apriori - association rule induction / frequent item set mining. http://www.borgelt.net/apriori.html.

[9] Stephan A. Brandt and Lawrence W. Stark. Spontaneous Eye Movements During Visual Imagery Reflect the Content of the Visual Scene. *J. Cognitive Neuroscience*, 9(1):27–38, January 1997.

[10] Andy Brown, Caroline Jay, and Simon Harper. Tailored Presentation of Dynamic Web Content for Audio Browsers. *International Journal of Human-Computer Studies*, 70(3):179 – 196, 2012.

[11] M. Burmester and M. Mast. Repeated Web Page Visits and the Scanpath Theory: A Recurrent Pattern Detection Approach. *Journal of Eye Movement Research*, 3(4):1–20, 2010.

[12] Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. Accordion Summarization for End-game Browsing on PDAs and Cellular Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 213–220, New York, NY, USA, 2001. ACM.

[13] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma. VIPS: a Vision Based Page Segmentation Algorithm. Technical report, Microsoft Research, 2003. Technical Report MSR-TR-2003-79.

[14] Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 225–233, New York, NY, USA, 2003. ACM.

[15] Chung Han Chiang. A Genetic Algorithm for the Longest Common Subsequence of Multiple Sequences. Master's thesis, National Sun Yat-sen University, 2009.

[16] Filipe Cristino, Sebastiaan Mathot, Jan Theeuwes, and Iain D. Gilchrist. ScanMatch: a Novel Method for Comparing Fixation Sequences. *Behav Res Methods*, 42(3):692–700, 2010.

[17] Edward Cutrell and Zhiwei Guan. What are You Looking for?: An Eye-tracking Study of Information Usage in Web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 407–416, New York, NY, USA, 2007. ACM.

[18] Rong-Fuh Day. Examining the Validity of the Needleman and Wunsch Algorithm in Identifying Decision Strategy with Eye-movement Data. *Decision Support Systems*, 49(4):396 – 403, 2010.

[19] Gautier Drusch and J. M. Christian Bastien. Analyzing Web Pages Visual Scanpaths: Between and WithinTasks Variability. *Work: A Journal of Prevention, Assessment and Rehabilitation*, 41:1559–1566, 2012.

[20] Claudia Ehmke and Stephanie Wilson. Identifying Web Usability Problems from Eye-tracking Data. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it*, volume 1 of *BCS-HCI '07*, pages 119–128, Swinton, UK, 2007. British Computer Society.

[21] Tom Foulsham and Geoffrey Underwood. What can Saliency Models Pedict about Eye Movements? Spatial and Sequential Aspects of Fixations during Encoding and Recognition. *Journal of Vision*, 8(2):1–17, 2008.

[22] Joystone Gbadamosi and Wolfgang H. Zangemeister. Visual Imagery in Hemianopic Patients. *Journal of Cognitive Neuroscience*, 13(7):855–866, October 2001.

[23] Joseph H. Goldberg and Jonathan I. Helfman. Scanpath Clustering and Aggregation. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications*, ETRA '10, pages 227–234, New York, NY, USA, 2010. ACM.

[24] Joseph H. Goldberg and Xerxes P. Kotval. Computer Interface Evaluation using Eye Movements: Methods and Constructs. *International Journal of Industrial Ergonomics*, 24(6):631–645, October 1999.

[25] Joseph H. Goldberg, Mark J. Stimson, Marion Lewenstein, Neil Scott, and Anna M. Wichansky. Eye Tracking in Web Search Tasks: Design Implications. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, ETRA '02, pages 51–58, New York, NY, USA, 2002. ACM.

[26] GSM Association. The Mobile Economy 2013. http://www.gsmamobileeconomy.com/, 2013.

[27] Zhiwei Guan and Edward Cutrell. An Eye Tracking Study of the Effect of Target Rank on Web Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 417–420, New York, NY, USA, 2007. ACM.

[28] Helene Hembrooke, Matt Feusner, and Geri Gay. Averaging Scan Patterns and What They Can Tell Us. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA '06, pages 41–41, New York, NY, USA, 2006. ACM.

[29] John Heminghous and Andrew T. Duchowski. iComp: A Tool for Scanpath Visualization and Comparison. In *Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, APGV '06, pages 152–152, New York, NY, USA, 2006. ACM.

[30] John M. Henderson, James R. Brockmole, Monica S. Castelhano, and Michael Mack. Visual Saliency Does Not Account for Eye Movements during Visual Search in Real-World Scenes. In Roger Van Gompel, Martin Fischer, Wayne Murray, and Robin Hill, editors, *Eye Movement Research: Insights into Mind and Brain*, chapter 25, pages 537–562. Elsevier, 2006.

[31] Gord Hotchkiss, Steve Alston, and Greg Edwards. Eye Tracking Study: An In Depth Look at Interactions with Google using Eye Tracking Methodology, June 2005.

[32] Kuosi Huang, Chang biau Yang, and Kuo-Tsung Tseng. Fast Algorithms for Finding the Common Subsequence of Multiple Sequences. In *Proceedings of the International Computer Symposium*, 2004.

[33] Laurent Itti and Christof Koch. A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention. *Vision Research*, 40:1489–1506, 2000.

[34] Laurent Itti, Christof Koch, and Ernst Niebur. A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, Nov 1998.

[35] Robert J. K. Jacob and Keith S. Karn. Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. *The Mind's eye: Cognitive The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, 2003:573–603, 2003.

[36] Halszka Jarodzka, Kenneth Holmqvist, and Marcus Nyström. A Vector-based, Multidimensional Scanpath Similarity Measure. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications*, ETRA '10, pages 211–218, New York, NY, USA, 2010. ACM.

[37] Tao Jiang and Ming Li. On the Approximation of Shortest Common Supersequencesand Longest Common Subsequences. *SIAM J. Comput.*, 24(5):1122–1139, October 1995.

[38] Sheree Josephson and Michael E. Holmes. Visual Attention to Repeated Internet Images: Testing the Scanpath Theory on the World Wide Web. In *Proceedings of the 2002 symposium on Eye tracking research and applications*, ETRA '02, pages 43–49, New York, NY, USA, 2002. ACM.

[39] Chichang Jou. A Semantics-Based Automatic Web Content Adaptation Framework for Mobile Devices. In Joaquim Filipe and Jos Cordeiro, editors, *Web Information Systems and Technologies*, volume 8 of *Lecture Notes in Business Information Processing*, pages 230–242. Springer Berlin Heidelberg, 2008.

[40] Kristian Kersting, Luc De Raedt, and Bernd Gutmann. Rational Sequence Learning. In Luc De Raedt, Paolo Frasvconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming*, number LNAI 4911 in Theory and Applications, pages 28–55. Springer-Verlag, Berlin, 2008.

[41] Kerstin Klckner, Nadine Wirschum, and Anthony Jameson. Depth- and Breadth-First Processing of Search Result Lists. In *In CHI '04 Abstracts*, page 1539. ACM Press, 2004.

[42] Peter Krusche and Alexander Tiskin. Computing Alignment Plots Efficiently. In Barbara Chapman, Frederic Desprez, Gerhard R. Joubert, Alain Lichnewsky, Frans Peters, and Thierry Priol, editors, *Advances in Parallel Computing*, volume 19 of *Parallel Computing: From Multicores and GPU's to Petascale*, page pp. 158165. IOS Press, Amsterdam, 2010.

[43] Vladimir I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[44] Darren Lunn, Simon Harper, and Sean Bechhofer. Identifying Behavioral Strategies of Visually Impaired Users to Improve Access to Web Content. *ACM Trans. Access. Comput.*, 3(4):13:1–13:35, April 2011.

[45] Magnus S. Magnusson. Discovering Hidden Time Patterns in Behavior: T-patterns and Their Detection. *Behav Research Methods Instruments Computers*, 32(1):93–110, 2000.

[46] Mehregan Mahdavi, Hamid Khordadi, and Mohammad-Hassan Khoobkar. Web Transcoding for Mobile Devices Using a Tag-Based Technique. *World Applied Sciences Journal (Special Issue of Computer & Electrical Engineering)*, 10:49–58, 2010.

[47] S. Mannan, K. H. Ruddock, and D. S. Wooding. Automatic Control of Saccadic Eye Movements Made in Visual Inspection of Briefly Presented 2-D images. *Spatial Vision*, 9(3):363–86., 1995.

[48] Andrey Markov. Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. In R. Howard, editor, *Dynamic Probabilistic Systems (Volume I: Markov Models)*, chapter Appendix B, pages 552–577. John Wiley & Sons, Inc., New York City, 1971.

[49] Marcus Mast and Michael Burmester. Exposing Repetitive Scanning in Eye Movement Sequences with T-pattern Detection. In *Proceedings IADIS International conference interfaces and human computer interaction (IHCI)*, pages 137–145, Rome, Italy, 2011.

[50] Sebastiaan Mathot, Filipe Cristino, Iain D. Gilchrist, and Jan Theeuwes. A Simple Way to Estimate Similarity Between Pairs of Eye movement Sequences. *Journal of Eye Movement Research*, 5(1):115, 2012.

[51] John D Mccarthy, M Angela Sasse, and Jens Riegelsberger. Could I Have the Menu Please? An Eye Tracking Study of Design Conventions. In *In Proceedings of HCI2003*, pages 401–414. Springer-Verlag, 2003.

[52] Saul B. Needleman and Christian D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.

[53] Jakob Nielsen. F-Shaped Pattern for Reading Web Content. http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content. Nielsen Norman Group, April 2006.

[54] Jakob Nielsen and Kara Pernice. *Eyetracking Web Usability*. New Riders, Berkeley, CA, 2010.

[55] Bing Pan, Helene A. Hembrooke, Geri K. Gay, Laura A. Granka, Matthew K. Feusner, and Jill K. Newman. The Determinants of Web Page Viewing Behavior: An Eye-tracking Study. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, ETRA '04, pages 147–154, New York, NY, USA, 2004. ACM.

[56] Jian Pei, Jiawei Han, Senior Member, Behzad Mortazavi-asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions on Knowledge and Data Engineering*, 16:2004, 2004.

[57] Alex Poole and Linden J. Ball. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future. In *Prospects, Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc*, 2005.

[58] Alex Poole, LindenJ Ball, and Peter Phillips. In Search of Salience: A Response-time and Eye-movement Analysis of Bookmark Recognition. In Sally Fincher, Panos Markopoulos, David Moore, and Roy Ruddle, editors, *People and Computers XVIII-Design for Life*, pages 363–378. Springer London, 2005.

[59] Arun K Pujari. *Data Mining Techniques*. Hyderguda: Universities Press, 8th edition, 2005.

[60] Kari-Jouko Raiha. Some Applications of String Algorithms in Human-Computer Interaction. In Tapio Elomaa, Heikki Mannila, and Pekka Orponen, editors, *Algorithms and Applications*, volume 6060 of *Lecture Notes in Computer Science*, pages 196–209. Springer Berlin Heidelberg, 2010.

[61] Pia Rama and Thierry Baccino. Eye Fixationrelated Potentials (EFRPs) during Object Identification. *Visual Neuroscience*, 27(doi:10.1017/S0952523810000283):pp 187–192, 2010.

[62] Keith Rayner. Eye Movements in Reading and Information Processing: 20 Years of Research. *Psychological Bulletin*, 124:372–422, 1998.

[63] Keith Rayner and Alexander Pollatsek. *The Psychology of Reading*. Englewood Cliffs: Prentice Hall, 1989.

[64] Keith Rayner, Tim J Smith, George L Malcolm, and John M Henderson. Eye Movements and Visual Encoding During Scene Perception. *Psychological science*, 20:6–10, 2009.

[65] James A. Renshaw, Janet Finlay, David A. Tyfa, and Robert D. Ward. Designing for Visual Influence: An Eye Tracking Study of the Usability of Graphical Management Information. In *IFIP Conference on Human-Computer Interaction*, 2003.

[66] W3C WAI Research and Development Working Group (RDWG). Research Report on Mobile Web Accessibility. In Simon Harper, Peter Thiessen, and Yeliz Yesilada, editors, *W3C WAI Symposium on Mobile Web Accessibility*, W3C WAI Research and Development Working Group (RDWG) Notes. W3C Web Accessibility Initiative (WAI), first public working draft edition, December 2012.

[67] Dario D. Salvucci and Joseph H. Goldberg. Identifying Fixations and Saccades in Eye-tracking Protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM.

[68] Anthony Santella and Doug DeCarlo. Robust Clustering of Eye Movement Recordings for Quantification of Visual Interest. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, ETRA '04, pages 27–34, New York, NY, USA, 2004. ACM.

[69] Peter H. Sellers. On the Theory and Computation of Evolutionary Distances. *SIAM Journal on Applied Mathematics*, 26(4):787–793, 1974.

[70] S. C. Sharma. *Operation Research: Pert, Cpm and Cost Analysis*. Discovery Publishing House: New Delhi, 2006.

[71] Steven S. Skiena. *The Algorithm Design Manual*. London: Springer-Verlag, second edition edition, 2008.

[72] Temple F. Smith and Michael S. Waterman. Identification of Common Molecular Subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[73] SR Research Ltd. *EyeLink II User Manual*, February 2006.

[74] Hironobu Takagi, Shin Saito, Kentarou Fukuda, and Chieko Asakawa. Analysis of Navigability of Web Applications for Improving Blind Usability. *ACM Trans. Comput.-Hum. Interact.*, 14(3):13, September 2007.

[75] Haruhiko Takeuchi and Yoshiko Habuchi. A Quantitative Method for Analyzing Scan Path Data Obtained by Eye Tracker. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 283 –286, 1 2007-april 5 2007.

[76] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[77] Tobii Technology. Tobii Eye Tracking: An introduction to Eye Tracking and Tobii Eye Trackers. Technical report, TobiiTechnology, 2010.

[78] Tobii Technology AB. *Tobii Studio 2.X User Manual (version September 2010)*, 2010.

[79] Hoi Ying Tsang, Melanie Tory, and Colin Swindells. eSeeTrack: Visualizing Sequential Fixation Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):953–962, November 2010.

[80] Robert A. Wagner and Michael J. Fischer. The String-to-String Correction Problem. *J. ACM*, 21(1):168–173, January 1974.

[81] Jianyong Wang and Jiawei Han. BIDE: Efficient Mining of Frequent Closed Sequences. In *Proceedings of the 20th International Conference on Data Engineering*, ICDE '04, pages 79–, Washington, DC, USA, 2004. IEEE Computer Society.

[82] Julia M. West, Anne R. Haake, Evelyn P. Rozanski, and Keith S. Karn. eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA '06, pages 149–154, New York, NY, USA, 2006. ACM.

[83] Xiangye Xiao, Qiong Luo, Dan Hong, Hongbo Fu, Xing Xie, and Wei-Ying Ma. Browsing on Small Displays by Transforming Web pages into Hierarchically Structured Subpages. *ACM Trans. Web*, 3(1):4:1–4:36, January 2009.

[84] Christopher C. Yang and Fu Lee Wang. Fractal Summarization for Mobile Devices to Access Large Documents on the Web. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 215–224, New York, NY, USA, 2003. ACM.

[85] Yeliz Yesilada, Alan Chuter, and Shawn Lawton Henry. Shared Web Experiences: Barriers Common to Mobile Device Users and People with Disabilities, 2008.

[86] Yeliz Yesilada, Simon Harper, Carole Goble, and Robert Stevens. Screen Readers Cannot See. In *Web Engineering, Proceedings*, pages 445–458. Springer, 2004.

[87] Yeliz Yesilada, Robert Stevens, Simon Harper, and Carole Goble. Evaluating DANTE: Semantic Transcoding for Visually Disabled Users. *ACM Trans. Comput.-Hum. Interact.*, 14(3):14, September 2007.

[88] Xinyi Yin and Wee Sun Lee. Using Link Analysis to Improve Layout on Mobile Devices. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 338–344, New York, NY, USA, 2004. ACM.

[89] Gregory J. Zelinsky and David L. Sheinberg. Eye Movements during Parallel-serial Visual Search. *Journal of Experimental Psychology: Human Perception and Performance*, 23:244–262., 1997.