Identifying Patterns in Eyetracking Scanpaths in Terms of Visual Elements of Web Pages

Sukru Eraslan[1,2], Yeliz Yesilada[1], and Simon Harper[2]

[1] Middle East Technical University, Northern Cyprus Campus, Guzelyurt, Mersin 10, Turkey
{seraslan,yyeliz}@metu.edu.tr

[2] University of Manchester, School of Computer Science, United Kingdom
sukru.eraslan@postgrad.manchester.ac.uk,
simon.harper@manchester.ac.uk

Abstract. Web pages are typically decorated with different kinds of visual elements that help sighted people complete their tasks. Unfortunately, this is not the case for people accessing web pages in constraint environments such as visually disabled or small screen device users. In our previous work, we show that tracking the eye movements of sighted users provide good understanding of how people use these visual elements. We also show that people's experience in constraint environments can be improved by reengineering web pages by using these visual elements. However, in order to reengineer web pages based on eyetracking, we first need to aggregate, analyse and understand how a group of people's eyetracking data can be combined to create a common scanpath (namely, eye movement sequence) in terms of visual elements. This paper presents an algorithm that aims to achieve this. This algorithm was developed iteratively and experimentally evaluated with an eyetracking study. This study shows that the proposed algorithm is able to identify patterns in eyetracking scanpaths and it is fairly scalable. This study also shows that this algorithm can be improved by considering different techniques for pre-processing the data, by addressing the drawbacks of using the hierarchical structure and by taking into account the underlying cognitive processes.

Keywords: eyetracking, scanpaths, commonality, transcoding, reengineering.

1 Introduction

Web pages mainly consist of different kinds of visual elements, such as menu, logo and hyperlinks. These visual elements help sighted people complete their tasks, but unfortunately small screen device users and disabled users cannot benefit from these elements. When people access web pages with small screen devices, they typically experience many difficulties [1]. For example, on small screen devices, only some parts of web pages are accessible or the complete web page is available with very small text size. Hence, they may need to scroll or zoom a lot which can be annoying. Moreover, they may need more time and effort to find their targets. Similarly, web experience can be challenging for visually disabled users who typically use screen readers to access the web [2]. Since screen readers follow the source code of web pages, visually disabled users have to listen to unnecessary clutter to get to the main content [3].

S. Casteleyn, G. Rossi, and M. Winckler (Eds.): ICWE 2014, LNCS 8541, pp. 163–180, 2014.

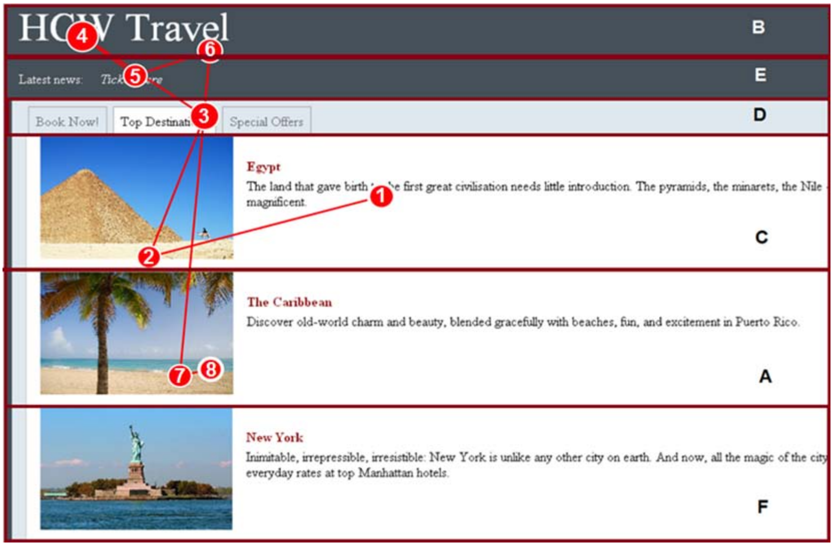

Fig. 1. A scanpath on a segmented web page

In our previous work, we show that reengineering web pages by using the visual elements can improve the user experience in constraint environments [4]. However, identifying visual elements and their role is the key for such reengineering process. To automatically process a web page and identify these elements, in our previous work we have extended and improved the Vision Based Page Segmentation (VIPS) algorithm [5,6]. This extended algorithm automatically discovers visual elements and relates them to the underlying source code. It allows direct access to these visual elements via XPath. However, this algorithm does not provide any information on how these visual elements are used. In our previous work, we also show that tracking the eye movements of sighted users provide good understanding of how they are used [2]. Eyes make quick movements which are called saccades. Between saccades, eyes make fixations where they become relatively stationary. Both fixations and saccades create scanpaths which are eye movement sequences [7]. Fig. 1 shows how a web page is segmented and illustrates a scanpath on a segmented web page. The circles represent fixations where the larger circles represent longer fixations. The numbers in the circles show the sequence. Also, the lines between circles are saccades.

In order to be able to use eyetracking data for reengineering web pages, this paper presents an algorithm called "*eMine scanpath algorithm*"[1]. This algorithm analyses and aggregates a group of people's eyetracking data to create a common scanpath in terms of visual elements of web pages (Section 3). Web pages are first automatically segmented into visual elements with the extended and improved version of the VIPS algorithm [6,5]. Eyetracking data is then exported and related to these visual elements. This creates individual scanpaths of users in terms of visual elements. These individual

[1] `http://emine.ncc.metu.edu.tr/`

scanpaths are then used by eMine scanpath algorithm to create a common scanpath. eMine scanpath algorithm was iteratively developed with the existing eyetracking data and our preliminary evaluation of this algorithm with the existing data was promising [8]. But in order to experientially evaluate validity and scalability of this algorithm, we conducted a new eyetracking study with 40 participants (Section 4). This study illustrates that eMine scanpath algorithm is able to identify a common scanpath in terms of visual elements of web pages and it is fairly scalable (Section 5 and Section 6). It has also revealed some weaknesses which can be improved in the future (Section 7).

2 Related Work

Eyetracking scanpaths have been analysed with different methods for different purposes. These methods typically use string representations of scanpaths which are generated using the sequence of Areas of Interest (AoIs) [9]. For example, the string representation of the scanpath in Fig. 1 is generated as CCDBEBAA. Different ways can be used to generate these AoIs such as using a grid layout directly [9] or the fixations' distribution over web pages [10]. However, these existing approaches typically treat a web page as an image to identify these AoIs which means these scanpaths cannot be used to process web pages. In order to address this, our previous work automatically segments a web page and each segment becomes an AoI [5,6]. This allows relating AoIs with the underlying source code which is important for being able to process web pages by using the eyetracking data.

The Levenshtein Distance (String-Edit) algorithm has commonly been used to analyse scanpaths [11,9]. This algorithm calculates the dissimilarity between the string representations of two scanpaths by transforming one to another with a minimum number of operations (insertion, deletion and substitution). For example, the dissimilarity between XYCZ and XYSZ is calculated as 1 (one) by the String-Edit algorithm because the substitution C with S is sufficient to transform one to another. Although the String-edit algorithm can be used to categorise scanpaths [12] and investigate differences between the behaviours of people on web pages [11], the algorithm itself is not able to identify a common scanpath for multiple scanpaths.

Transition Matrix is one of the methods which use multiple scanpaths to create a matrix [12]. This matrix allows identifying the possible next and previous AoI of the particular AoI. However, when this method is considered for identifying a common scanpath, some considerable problems arise, such as What is the start and end point of the common scanpath? Which probabilities should be considered?

To address these problems, some other methods can be considered. For example, the Shortest Common Supersequence method has been mentioned in literature to identify a common scanpath for multiple people but it has considerable weaknesses [13]. For example, it identifies XABCDEZ as a common scanpath for the individual scanpaths XAT, XBZ, XCZ, XDZ and XEZ. As can be easily recognised, the common scanpath is not supported by the individual scanpaths, for instance, the common scanpath has E which is included by only one individual scanpath (XEZ). Furthermore, the common scanpath is quite longer compared to the individual scanpaths.

Some methods, such as T-Pattern [14] and eyePatterns's Discover Patterns [12], have been proposed to detect subpatterns in eyetracking scanpaths. However, eyePatterns's Discover Patterns method [12] is not tolerant of extra items in scanpaths. For instance, XYZ can be detected as a subpattern for XYZ and WXYZ but it cannot be detected for XYZ and WXUYZ because of the extra item U. This shows that this method is reductionist which means it is likely to produce unacceptable short scanpaths.

The Multiple Sequence Alignment method was proposed to identify a common scanpath but this method was not validated [15]. Moreover, the Dotplots-based algorithm was proposed to identify a common scanpath for multiple people [16]. This algorithm creates a hierarchical structure by combining a pair of scanpaths with the Dotplots algorithm. The individual scanpaths are located at leafs whereas the common scanpath is located at the root. Some statistical methods have been applied to address the reductionist approach of the Dotplots algorithm [16].

We are interested in common patterns in eyetracking data instead of individual patterns to be able to reengineer web pages. However, as can be seen above, there is not much research in identifying common scanpaths and the existing ones are likely to produce unacceptable short common scanpaths. In this paper, we present our eMine scanpath algorithm to address the limitations of these existing approaches, especially the problem of being reductionist.

3 eMine Scanpath Algorithm

Algorithm 1 shows our proposed eMine scanpath algorithm [8] which takes a list of scanpaths and returns 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 scanpath, then it tries to find the most similar two scanpaths in the list by using the String-edit algorithm [11]. It then removes these two scanpaths from the list of scanpaths and introduces their common scanpath produced by the Longest Common Subsequence method [17] to the list of scanpaths. This continues until there is only one scanpath.

Algorithm 1. Find common scanpath

Input: Scanpath List
Output: Scanpath

```
if the size of Scanpath List is equal to 1 then
    return the scanpath in Scanpath List
end if
while the size of Scanpath List is not equal to 1 do
    Find the two most similar scanpaths in Scanpath List with the String-edit algorithm
    Find the common scanpath by using the Longest Common Subsequence method
    Remove the similar scanpaths from the Scanpath List
    Add the common scanpath to the Scanpath List
end while
return the scanpath in Scanpath List
```

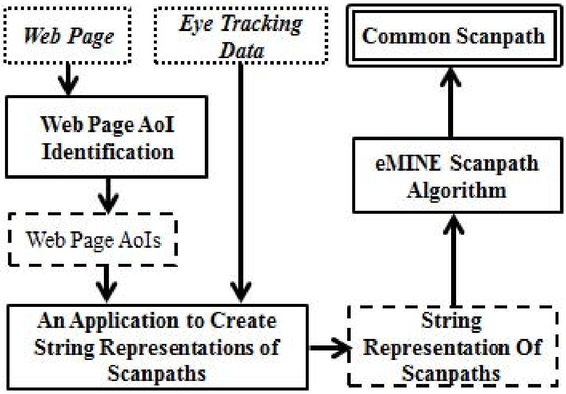

Fig. 2. System architecture where '...' shows the input parts, '_ _ _' represents intermediate parts, '_' illustrates the functional parts and '=' is used for the output part

3.1 System Architecture and Implementation

eMine scanpath algorithm was integrated with the extended and improved version of the VIPS algorithm [6,5]. Fig. 2 illustrates the system architecture which consists of the following parts: two input parts (web page and eyetracking data), three functional parts (web page AoI identification, an application to create string representations of scanpaths, eMine scanpath algorithm), two intermediate parts which are created as an output of one functional part and used as an input for another functional part (web page AoIs, string representations of scanpaths) and one output part (common scanpath). The functional parts are explained below.

Web Page AoI Identification. A web page is used as an input for the web page AoI identification part. This part creates AoIs automatically by using the extended and improved version of the VIPS algorithm [6,5]. Even though, the extended VIPS was used, it would be easily replaced by an alternative method of AoI identification approach. These AoIs represent visual elements of web pages.

An Application to Create String Representations of Scanpaths. The automatically generated web page AoIs and eyetracking data, provided by eyetracking software, are then used by an application to create string representations of scanpaths.

eMine Scanpath Algorithm. Once the string representations are created, our scanpath algorithm is applied to them to produce a common scanpath in terms of AoIs.

eMine scanpath algorithm[2] was implemented on the Accessibility Tools Framework (ACTF)[3] which is an open-source Eclipse project.

[2] `http://emine.ncc.metu.edu.tr/software.html`
[3] `http://www.eclipse.org/actf/`

4 An Eyetracking Study

In order to experimentally evaluate validity and scalability of eMine scanpath algorithm, we conducted an eyetracking study. This study aims to investigate the following two research questions:

1. *Validity:* The aim is to investigate whether or not eMine scanpath algorithm can successfully identify common scanpaths in terms of visual elements of web pages. Thus, we ask *"Can eMine algorithm identify common scanpaths in terms of visual elements of web pages?"*.
2. *Scalability:* We would like to investigate whether or not eMine scanpath algorithm works well for different numbers of participants on different web pages. Hence, the research question here is *"How does the number of individual scanpaths affect common scanpaths?"*.

4.1 Equipment

Participants sat in front of a 17" monitor with a built-in TOBII T60 eye tracker with screen resolution 1280 x 1024. The web pages were on a HP ELiteBook 8530p laptop and these web pages were shown to the participants using the eye tracker's screen. Tobii Studio eye gaze analysis software was used to record the data. Eyetracking data was also stored on that laptop, too. The collected eyetracking data were analysed on a 17" monitor with the screen resolution 1280 x 1024.

4.2 Materials

Six web pages were randomly selected from a group of pages that were used in our previous study. That study focused on evaluating the extended and improved version of the VIPS algorithm and to have continuity in our studies we used same set of pages [6,5]. These web pages were categorised based on their complexity, which were low, medium and high [6,5,18]. Two web pages were chosen randomly from each level of complexity for our study. These pages with their complexity levels are as follow: Apple (Low), Babylon (Low), AVG (Medium), Yahoo (Medium), Godaddy (High) and BBC (High). Since the 5th segmentation granularity level was found as the most successful level with approximately 74% user satisfaction, we decided to use the 5th level for our experiments [6,5]. The segmented web pages can be seen in Fig. 3, 4, 5, 6, 7 and 8.

4.3 Procedure

This eyetracking study consists of the following three parts.

Introduction: The participants read the information sheet and signed the consent form. Next, they filled in the short questionnaire which was for the purpose of collecting basic demographic information of participants, which are gender, age groups and education level. The participants were also asked to rank their web page usage for the six web pages with 1 (Daily), 2 (Weekly), 3 (Monthly), 4 (Less than once a month) or 5 (Never).

Main Part: The participants sat in front of the eye tracker which calibrated to their gaze. They then viewed all of the six web pages twice, one view for searching (maximum 120 seconds) and one view for browsing in a random order. For browsing tasks, the participants were given 30 seconds as used in other studies [19]. The searching and browsing tasks are shown in Table 1. The researcher was responsible to check if the participants complete the tasks successfully and take notes if necessary.

Conclusion: At the end, the participants were asked to redraw three web pages from three different complexity levels.

4.4 User Tasks

User tasks are categorised into two groups for this study: searching and browsing. In the literature, many studies were conducted to categorise user tasks on the web [20]. G. Marchionini Search Activities Model is one of the most popular models in this field [20]. It consists of three groups which are lookup, learn and investigate [20]. Our searching category is related to fact finding which is associated with the lookup group whereas our browsing category is related to serendipitous browsing which is associated with the investigation group. The tasks which are defined for the six web pages are listed in Table 1.

We designed the system to ensure that half of the participants complete searching tasks firstly and then complete browsing tasks. Other half completed browsing task firstly and then completed searching tasks. The reason is to prevent familiarity effects on eye movements which can be caused by the user tasks.

4.5 Participants

The majority of the participants comprised students, along with some academic and administrative staff at Middle East Technical University Northern Cyprus Campus and the University of Manchester. Twenty male and twenty female volunteers participated. One male participant changed his body position during the study, so the eye tracker could not record his eye movements. Another male participant had no successful eye calibration. Unfortunately, these two participants were excluded from the study. Therefore, the eyetracking data of 18 males and 20 females were used to evaluate eMine scanpath algorithm.

All of the participants use the web daily. Most of the participants (18 participants) are aged between 18 and 24 years old, then 25-34 group (14 participants) and 35-54 group (6 participants). Moreover, 14 participants completed their high/secondary schools, 6 participants have a bachelor’s degree, 9 participants have a master’s degree and 9 participants completed their doctorate degrees.

5 Results

In this section, we present the major findings of this study in terms of the two research questions presented in Section 4.

Table 1. Tasks used in the eyetracking study

Apple	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you locate a link which allows watching the TV ads relating to iPad mini? 2. Can you locate a link labelled iPad on the main menu?
Babylon	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you locate a link you can download the free version of Babylon? 2. Can you find and read the names of other products of Babylon?
Yahoo	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you read the titles of the main headlines which have smaller images? 2. Can you read the first item under News title?
AVG	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you locate a link which you can download a free trial of AVG Internet Security 2013? 2. Can you locate a link which allows you to download AVG Antivirus FREE 2013?
GoDaddy	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you find a telephone number for technical support and read it? 2. Can you locate a text box where you can search a new domain?
BBC	
Browsing	1. Can you scan the web page if you find something interesting for you?
Searching	1. Can you read the first item of Sport News? 2. Can you locate the table that shows market data under Business title?

5.1 Validity

"Can eMine scanpath algorithm identify common scanpaths in terms of visual elements of web pages?"

The participants were asked to complete some searching tasks on web pages, therefore we are expecting to see that the common scanpath supports those tasks. We used eMine scanpath algorithm to identify a common scanpath for each of the six web pages. Some participants could not complete the searching tasks successfully and/or had calibration problems. These participants were defined as unsuccessful participants and excluded from the study. The success rates in completing searching tasks are as follow: Apple: 81.58 %, Babylon: 94.74 %, AVG: 94.74 %, Yahoo: 84.21 %, Godaddy: 73.68 % and BBC: 100 %. These values are calculated by dividing the number of the successful participants by the total number of the participants on the page.

Table 2 shows the common scanpaths and the abstracted common scanpaths produced by eMine scanpath algorithm for the web pages where 'P' represents the number of successful participants. In order to have abstracted common scanpaths, their string representations are simplified by abstracting consecutive repetitions [21,22]. For instance, MMPPQRSS becomes MPQRS.

Table 2. The common scanpaths produced by eMine scanpath algorithm for the web pages

Page Name	P	Common Scanpath	Abstracted Common Scanpath
Apple	31	EEB	EB
Babylon	36	MMPPQRSS	MPQRS
AVG	36	GGGGGGGGGGGGGGGIIIIIIIII	GI
Yahoo	32	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	I
Godaddy	28	OOOOMMMMMM	OM
BBC	38	RNNNNN	RN

On the Apple web page, 31 out of 38 participants were successful. On this page, the participants were asked to locate a link which allows watching the TV ads relating to iPad mini and then locate a main menu item iPad. EB is identified as a common scanpath for these participants. Since E is associated with the first part and B is related to the second part of the searching task, this common scanpath completely supports the searching task. Fig. 3 shows this common scanpath on the Apple web page.

Fig. 3. Common scanpath on the Apple web page

On the Babylon web page, only 2 participants out of 38 were not successful. On this page, the participants were requested to locate a link which allows downloading a free version of Babylon and then read the names of other products of Babylon. The common scanpath for the 36 participants was identified as MPQRS shown in Fig. 4. M is related with a free version of Babylon whereas P, Q, R and S are associated with four other products of Babylon. Therefore, the common scanpath thoroughly supports the searching task.

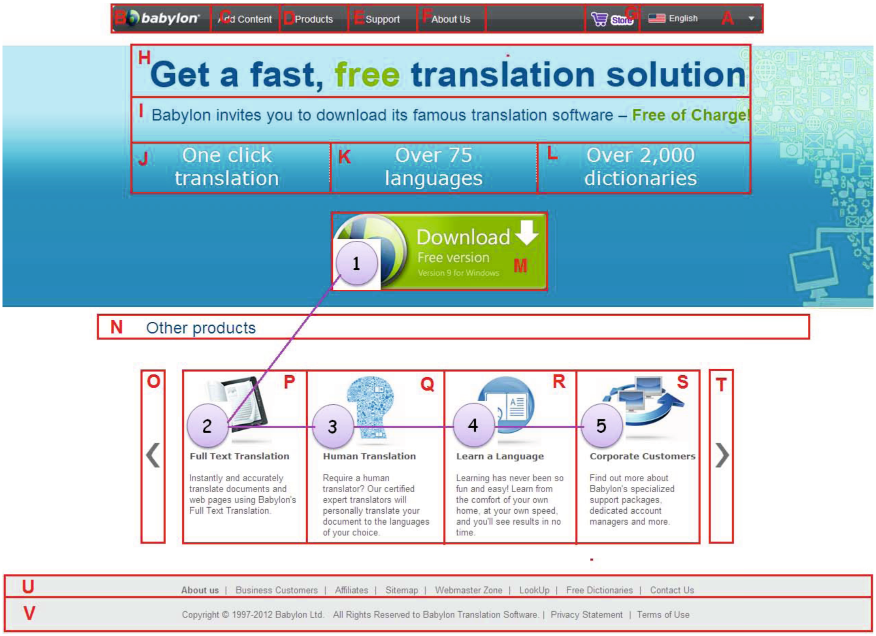

Fig. 4. Common scanpath on the Babylon web page

Similar to the Babylon web page, only 2 participants were unsuccessful on the AVG web page. The searching task here was locating a link which allows downloading a free trial of AVG Internet Security 2013 and then locating a link which allows downloading AVG Antivirus FREE 2013. The common scanpath was produced as GI where G has a link to download a free trial of AVG Internet Security 2013 and I contains a link to download AVG Antivirus FREE 2013. Therefore, the common scanpath, shown in Fig. 5, entirely supports the searching task.

For the Yahoo web page, 6 participants could not be successful. The participants required to read the titles of the main headlines which have smaller images and then read the first item under News title. Since only I is produced as a common scanpath on this web page and I contains both parts of the task, the common scanpath nicely supports the searching task, too. Fig. 6 shows this common scanpath.

Since 28 out of 38 participants were successful, 10 participants were excluded for the Godaddy web page. The successful participants read the telephone number for technical support and then located a text box where they can search for a new domain. eMine scanpath algorithm produced OM as a common scanpath shown in Fig. 7. Since M

Fig. 5. Common scanpath on the AVG web page

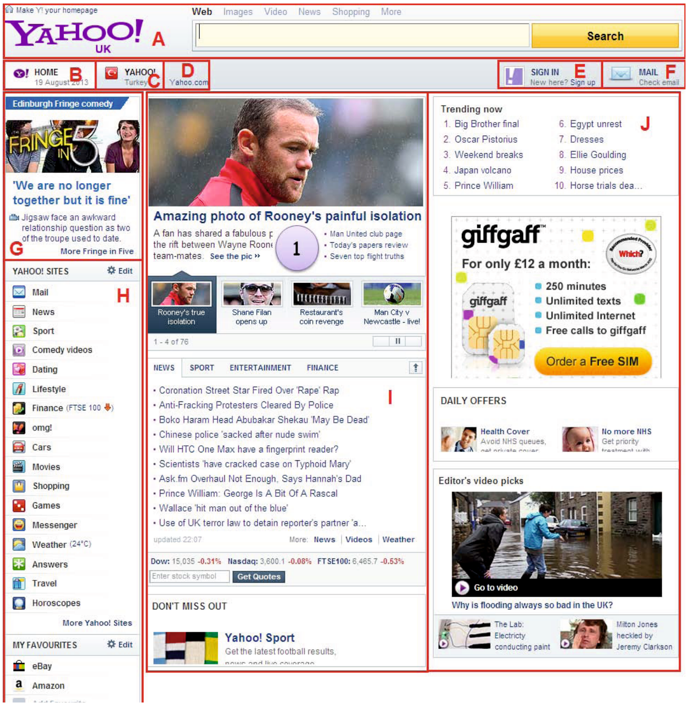

Fig. 6. Common scanpath on the Yahoo web page

Fig. 7. Common scanpath on the Godaddy web page

Fig. 8. Common scanpath on the BBC web page

contains the text box and there is no AoI in the scanpath which is related with the telephone number, the common scanpath partially supports the searching task on the Godaddy web page.

On the BBC web page, all participants completed the searching task successfully. The participants were asked to read the first item of the sports news and then locate a table which shows the market data. Therefore, the participants needed to locate R and then N. As the common scanpath RN is produced, it supports the searching task very well. Fig. 8 illustrates this common scanpath on the BBC web page.

To sum up, the common scanpaths on the Apple, Babylon, AVG, Yahoo and BBC web pages completely support the searching tasks whereas the common scanpath on the Godaddy web page partially supports the searching task.

5.2 Scalability

"How does the number of individual scanpaths affect common scanpaths?"

In order to test whether or not eMine scanpath algorithm works well with different numbers of individual scanpaths, we tested the algorithm with different numbers of individual participants. The participants were selected randomly from all of the successful participants. Table 3 illustrates the common scanpaths in terms of AoIs on the different web pages for 10, 20, 30 and 30+ participants while browsing and searching.

Table 3. The common scanpaths on the different web pages for 10, 20, 30 and 30+ participants while browsing and searching where '-' means that there was no sufficient number of successful participants and '—–' means that no common scanpath was detected

Task	Page Name	P=10	P=20	P=30	P=30+
Browsing	Apple	IF	F	F	F
	Babylon	MS	M	M	M
	AVG	GIG	G	G	G
	Yahoo	IJI	I	I	I
	Godaddy	O	O	O	O
	BBC	LP	LP	P	—–
Searching	Apple	EB	EB	EB	EB
	Babylon	MPQRS	MPQRS	MPQRS	MPQRS
	AVG	IGI	GI	GI	GI
	Yahoo	I	I	I	I
	Godaddy	OM	OM	-	-
	BBC	LPRN	RN	RN	RN

In order to see how the common scanpaths are affected when the number of participants increases, we calculated the similarities between the scanpaths which were produced for 10, 20, 30 and 30+ participants. To calculate the similarity between two common scanpaths the String-edit distance between two common scanpaths is divided by the length of the longer common scanpath to have a normalised score [23]. The purpose of a normalised score is to prevent any inconsistencies in similarities caused by different lengths [23,24]. Finally, the normalised score is subtracted from 1 [23]. For example, the common scanpath for 10 participants is LPRN and the common scanpath for 20 participants is RN on the BBC web page for the searching task. The String-edit distance is calculated as 2 between two scanpaths. After that, since the length of the longer scanpath (LPRN) is equal to 4, this distance is divided by 4. As a result, the normalised score is equal to 0.5. To calculate the similarity 0.5 is subtracted from 1, so the similarity between the two common scanpaths is equal to 0.5 (50 %). Table 4 shows these similarities between the common scanpaths for the searching task on the BBC web page whereas Table 5 illustrates the similarities between the common scanpaths for the browsing task on the Yahoo web page as examples.

Table 4. The similarities between the common scanpaths on the BBC web page for 10, 20, 30 and 30+ participants while searching

BBC Searching	P = 10	P = 20	P = 30	P = 30+
P = 10	—	50	50	50
P = 20	50	—	100	100
P = 30	50	100	—	100
P = 30+	50	100	100	—

Table 5. The similarities between the common scanpaths on the Yahoo web page for 10, 20, 30 and 30+ participants while browsing

Yahoo Searching	P = 10	P = 20	P = 30	P = 30+
P = 10	—	33.3	33.3	33.3
P = 20	33.3	—	100	100
P = 30	33.3	100	—	100
P = 30+	33.3	100	100	—

For both the browsing and searching tasks, we calculated the average similarity between the common scanpaths on each web page. To calculate these average similarities we divided the sum of the similarities between the scanpaths for 10, 20, 30 and 30+ participants by the total number of the similarities. In addition, we calculated the average similarity for both the browsing and searching tasks. Since each web page typically has four scanpaths (for 10, 20, 30 and 30+ participants), we determined their weights based on the number of scanpaths. All of the pages' weights are set to 4, except the Godaddy page because of the searching task. The Godaddy page has one common scanpath for 10 participants and one common scanpath for 20 participants, therefore its weight is set to 2. When the average is calculated, we multiplied the value with its weight to find the weighted value. After that, we found the sum of the weighted value and divided it by the sum of the weights. It was found that the average similarity for searching tasks (92.42%) is higher than the average similarity for the browsing task (69.44 %).

Table 6. The average of the similarities between the common scanpaths on each web page for 10, 20, 30 and 30+ participants

Page Name	Task	Average Similarity for Each Page
Apple	Browsing	75
Babylon	Browsing	75
AVG	Browsing	66.65
Yahoo	Browsing	66.65
Godaddy	Browsing	100
BBC	Browsing	33.33
Average Similarity for the 6 Pages	Browsing	69.44
Apple	Searching	100
Babylon	Searching	100
AVG	Searching	83.3
Yahoo	Searching	100
Godaddy	Searching	100
BBC	Searching	75
Average Similarity for the 6 Pages	Searching	92.42

6 Discussion

The eMine scanpath algorithm was experimentally evaluated with an eyetracking study and this study illustrates that the algorithm is able to successfully identify common scanpaths in terms of visual elements of web pages and it is fairly scalable.

The searching tasks completed by the participants on the given pages were used to validate eMine scanpath algorithm. We expected that the common scanpaths should support these searching tasks. For instance, on the Babylon web page, the participants were asked to locate the link which allows downloading the free version of Babylon (related to AoI M) and then read the names of other products of Babylon (related to AoIs P, Q, R and S). Therefore, we expected that the common scanpath on the Babylon web page should involve at least MPQRS for the searching tasks.

The results in Section 5.1 show that the common scanpaths produced by eMine scanpath algorithm completely support these tasks, except the common scanpath on the Godaddy page. On that page, the participants were asked to read a telephone number for technical support and locate the text box where they can search for a new domain. The common scanpath involves the AoI for the text box but does not include the AoI for the telephone number. Thus, it partially supports the searching task. There may be various reasons: (1) The participants might make a very few fixations on that AoI (2) Some participants might find the telephone number directly whereas some of them looked at many AoIs to find the telephone number. Therefore, it would be good to pre-process eyetracking data in depth to investigate the individual differences and their reasons.

Some other methods could also be used to validate eMine scanpath algorithm. One might consider calculating the similarities between the individual scanpaths and the common scanpath. Besides, the AoIs appeared in all individual scanpaths might be detected and then one part of the validation process could be done by using these AoIs.

The scalability of eMine scanpath algorithm was tested by using the different numbers of individual scanpaths as mentioned in Section 5.2. As expected, we can see that the algorithm is more scalable with the searching tasks because the participants were asked to complete some specific searching tasks. The average similarity is equal to 92.42 % between the common scanpaths which were produced with the different number of scanpaths for the searching tasks. However, the average similarity is equal to 69.44 % for the browsing tasks. Based on these values we can suggest that our algorithm is fairly scalable, especially in searching tasks.

There are some differences between scanpaths, such as producing LPRN for 10 participants and RN for 30+ participants on the BBC page. It is caused by using the hierarchical structure. As mentioned in Section 3, eMine scanpath algorithm uses a hierarchical structure while identifying common scanpaths. It selects the two most similar scanpaths from the list and finds their longest common subsequence. It is iteratively repeated until a single scanpath left. Because of the hierarchical structure, some information in intermediate levels can be lost because of combining two scanpaths.

Assume that there are three sequences: S1: GATACCAT S2: CTAAAGTC and S3: GCTATTGCG [17]. S1 and S2 can be aligned firstly and then S1'= - - A - A - - A - - - can obtained [17]. Following this, S1' and S3 can be aligned and then S3'= - - - A - - - - - - - - can be obtained [17]. This example clearly illustrates that the hierarchical structure can make the method reductionist. Here, all of the three scanpaths have G and T in different locations but G and T do not exist at the end. This may cause some differences in common scanpaths. Because of this reason, eMine scanpath algorithm was not able to identify any common scanpath on the BBC page for the browsing task. When a number of individual scanpaths is increased, the different most similar scanpath pairs can be generated and this may affect common scanpaths. Although eMINE scanpath algorithm has some drawbacks because of the hierarchical structure, it still partly addresses the reductionist problem of the other existing approaches (See Section 2).

To address the drawbacks of using the hierarchical structure a constraint might be created to prevent losing the AoIs appeared in all individual scanpaths in intermediate levels. Alternatively, some statistical approaches can be used to sort these AoIs and then create a common scanpath for multiple people.

7 Concluding Remarks and Future Work

This paper presents an algorithm and its evaluation that identifies common scanpaths in terms of visual elements of web pages. These visual elements are first automatically generated with the extended and improved version of the VIPS algorithm [6,5]. Eyetracking data is then related to these visual elements and individual scanpaths are created in terms of these visual elements. This algorithm then uses these individual scanpaths and generates a common scanpath in terms of these visual elements. This common scanpath can be used for reengineering web pages to improve the user experience in constraint environments.

To our knowledge, there is no work on correlating scanpaths with visual elements of web pages and the underlying source code, and this work is novel from that perspective [6,5]. This paper also shows how the validity and scalability of eMine scanpath algorithm was demonstrated with an eyetracking study. The results clearly show that this algorithm is able to identify common scanpaths in terms of visual elements of web pages and it is fairly stable. This algorithm aims to address the reductionist problem that the other existing work has, but the results show that there is still room for improvement.

The eyetracking study also suggests some directions for future work. It indicates that the individual differences can affect the identification of patterns in eyetracking scanpaths. Thus, eyetracking data should be pre-processed to investigate the individual differences and their reasons. Since an eye tracker collects a large amount of data, pre-processing is also required to eliminate noisy data. It is important because noisy data are likely to decrease the commonality in scanpaths. Another benefit of pre-processing is to identify outliers which are potential to decrease the commonality, too.

Finally, as with the existing scanpath methods, eMine scanpath algorithm also tends to ignore the complexities of the underlying cognitive processes. However, when people follow a path to complete their tasks on web pages, there may be some reasons that affect their decisions. Underlying cognitive processes can be taken into account while identifying common scanpaths.

Acknowledgments. 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. We would also like to thank our participants for their time and effort.

References

1. W3C WAI Research and Development Working Group (RDWG): Research Report on Mobile Web Accessibility. In: Harper, S., Thiessen, P., Yesilada, Y., (eds.): W3C WAI Symposium on Mobile Web Accessibility. W3C WAI Research and Development Working Group (RDWG) Notes. First public working draft edn. W3C Web Accessibility Initiative (WAI) (December 2012)
2. Yesilada, Y., Jay, C., Stevens, R., Harper, S.: Validating the Use and Role of Visual Elements of Web Pages in Navigation with an Eye-tracking Study. In: The 17th international Conference on World Wide Web, WWW 2008, pp. 11–20. ACM, New York (2008)
3. Yesilada, Y., Stevens, R., Harper, S., Goble, C.: Evaluating DANTE: Semantic Transcoding for Visually Disabled Users. ACM Trans. Comput.-Hum. Interact. 14(3), 14 (2007)
4. Brown, A., Jay, C., Harper, S.: Audio access to calendars. In: W4A 2010, pp. 1–10. ACM, New York (2010)
5. Akpınar, M.E., Yesilada, Y.: Heuristic Role Detection of Visual Elements of Web Pages. In: Daniel, F., Dolog, P., Li, Q. (eds.) ICWE 2013. LNCS, vol. 7977, pp. 123–131. Springer, Heidelberg (2013)
6. Akpınar, M.E., Yesilada, Y.: Vision Based Page Segmentation Algorithm: Extended and Perceived Success. In: Sheng, Q.Z., Kjeldskov, J. (eds.) ICWE Workshops 2013. LNCS, vol. 8295, pp. 238–252. Springer, Heidelberg (2013)
7. Poole, A., Ball, L.J.: Eye tracking in human-computer interaction and usability research: Current status and future Prospects. In: Ghaoui, C. (ed.) Encyclopedia of Human-Computer Interaction. Idea Group, Inc., Pennsylvania (2005)
8. Yesilada, Y., Harper, S., Eraslan, S.: Experiential transcoding: An EyeTracking approach. In: W4A 2013, p. 30. ACM (2013)

9. Takeuchi, H., Habuchi, Y.: A quantitative method for analyzing scan path data obtained by eye tracker. In: CIDM 2007, April 1-5, pp. 283–286 (2007)
10. Santella, A., DeCarlo, D.: Robust clustering of eye movement recordings for quantification of visual interest. In: ETRA 2004, pp. 27–34. ACM, New York (2004)
11. Josephson, S., Holmes, M.E.: Visual attention to repeated internet images: testing the scanpath theory on the world wide web. In: ETRA 2002, pp. 43–49. ACM, NY (2002)
12. West, J.M., Haake, A.R., Rozanski, E.P., Karn, K.S.: EyePatterns: software for identifying patterns and similarities across fixation sequences. In: ETRA 2006, pp. 149–154. ACM, New York (2006)
13. Räihä, K.-J.: Some applications of string algorithms in human-computer interaction. In: Elomaa, T., Mannila, H., Orponen, P. (eds.) Ukkonen Festschrift 2010. LNCS, vol. 6060, pp. 196–209. Springer, Heidelberg (2010)
14. Mast, M., Burmeister, M.: Exposing repetitive scanning in eye movement sequences with t-pattern detection. In: IADIS IHCI 2011, Rome, Italy, pp. 137–145 (2011)
15. Hembrooke, H., Feusner, M., Gay, G.: Averaging scan patterns and what they can tell us. In: ETRA 2006, p. 41. ACM, New York (2006)
16. Goldberg, J.H., Helfman, J.I.: Scanpath clustering and aggregation. In: ETRA 2010, pp. 227–234. ACM, New York (2010)
17. Chiang, C.H.: A Genetic Algorithm for the Longest Common Subsequence of Multiple Sequences. Master's thesis, National Sun Yat-sen University (2009)
18. Michailidou, E.: ViCRAM: Visual Complexity Rankings and Accessibility Metrics. PhD thesis, University of Manchester (2010)
19. Jay, C., Brown, A.: User Review Document: Results of Initial Sighted and Visually Disabled User Investigations. Technical report, University of Manchester (2008)
20. Marchionini, G.: Exploratory Search: From Finding to Understanding. Commun. ACM 49(4), 41–46 (2006)
21. Brandt, S.A., Stark, L.W.: Spontaneous Eye Movements During Visual Imagery Reflect the Content of the Visual Scene. J. Cognitive Neuroscience 9(1), 27–38 (1997)
22. Jarodzka, H., Holmqvist, K., Nyström, M.: A Vector-based, Multidimensional Scanpath Similarity Measure. In: ETRA 2010, pp. 211–218. ACM, New York (2010)
23. Foulsham, T., Underwood, G.: What can Saliency Models Predict about Eye Movements? Spatial and Sequential Aspects of Fixations during Encoding and Recognition. Journal of Vision 8(2), 1–17 (2008)
24. Cristino, F., Mathot, S., Theeuwes, J., Gilchrist, I.D.: Scanmatch: a novel method for comparing fixation sequences. Behavior Research Methods 42(3), 692–700 (2010)