Evaluation of Visualisation of Scanpath Trend Analysis (ViSTA) Tool

Ş. ERASLAN, S. KARABULUT, M.C. ATALAY and Y. YEŞILADA

Abstract—**Eye tracking plays a key role in user behaviour understanding and usability studies. We previously proposed an algorithm called STA (Scanpath Trend Analysis) that analyses multiple individual scanpaths on a web page to discover their trending path in terms of the areas of interest (AOIs). This algorithm provides the most representative path of multiple users compared to other algorithms (i.e., provides the most similar path to individual scanpaths). However, its current implementation has no graphical user interface and provides a sequence of characters that represent AOIs. Some external modules should also be installed in advance to run it. In our previous work, we presented the first web-based visualisation tool for the STA algorithm called ViSTA along with its initial evaluation. This tool allows to visualise individual scanpaths on a particular web page with gaze plots, visually draw AOIs, apply the STA algorithm, and visualise the result of the algorithm. In this paper, we present the extended version of ViSTA with a follow up user evaluation. The first version of ViSTA uses the STA algorithm which identifies trending AOIs based on all individual scanpaths. However, the extended one uses the STA algorithm with the tolerance level parameter which means trending elements can be identified based on a subset of individual scanpaths for discovering a more representative path. Both of our initial and follow up evaluations show that the workload in terms of NASA Task Load Index (TLX) is lower with ViSTA compared to the current implementation of the STA algorithm.**

Index Terms—**Eye Tracking, Areas of Interest, Web Pages, Trending Path, Web-based, User Interface.**

ŞÜKRÜ ERASLAN, Middle East Technical University, Northern Cyprus Campus, 99738 Kalkanlı, Güzelyurt, Mersin 10, Turkey (e-mail: seraslan@metu.edu.tr).

http://orcid.org/0000-0002-9277-8375

SERKAN KARABULUT, Middle East Technical University, Northern Cyprus Campus, 99738 Kalkanlı, Güzelyurt, Mersin 10, Turkey (e-mail: karabulut.serkan@metu.edu.tr).

https://orcid.org/0000-0002-6694-7219

MEHMET CAN ATALAY, Middle East Technical University, Northern Cyprus Campus, 99738 Kalkanlı, Güzelyurt, Mersin 10, Turkey (e-mail: atalay.can@metu.edu.tr).

https://orcid.org/0000-0002-9029-9880

YELİZ YEŞİLADA, Middle East Technical University, Northern Cyprus Campus, 99738 Kalkanlı, Güzelyurt, Mersin 10, Turkey (e-mail: yyeliz@metu.edu.tr).

http://orcid.org/0000-0001-8501-0205

Manuscript received November 30, 2018; accepted October 24, 2019.
DOI: 10.17694/bajece.490601

I. Introduction

EYE TRACKING has been commonly used for better understanding of how users interact with web pages, especially which areas are frequently used and which paths are usually followed in terms of these areas to complete certain tasks. Based on the analysis of eye tracking data, web pages can be further processed to support users in constrained environments for completing their tasks. In particular, web pages can be re-engineered for these users to allow them to directly access the most frequently used areas without being distracted by other inappropriate areas. This is especially useful for visually disabled users who access the web with their screen readers and have to listen to the whole page from top to bottom [1].

While users are interacting with web pages, their eyes make quick movements called saccades. Their eyes also make fixations on certain points. The series of saccades and fixations show their scanpaths on these pages. There are different techniques to visualise individual scanpaths [2] (see Section 2). The most popular technique for scanpath visualisation is a gaze plot that represents saccades and fixations with straight lines and circles respectively. The radius of a circle is directly proportional to the duration of a fixation and each circle is numbered to show the sequence. Figure 1 shows an example of a gaze plot on the home page of the Babylon website.

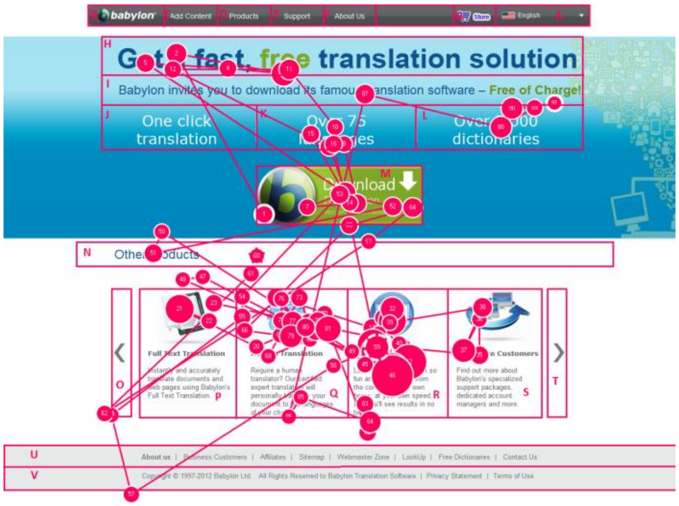

babylon
Add Content
Products
Support
About Us
English
H
Get fast, free translation solution
I
Babylon invites you to download its famous translation software – Free of Charge!
J
One click translation
K
Over 75 languages
L
Over 1,600 dictionaries
M
Download
N
Other products
O
P
Q
R
S
T
U
V
Full Text Translation
e Customers
About us | Business Customers | Affiliates | Sitemap | Webmaster Zone | LookUp | Free Dictionaries | Contact Us
Copyright © 1997-2012 Babylon Ltd. All Rights Reserved to Babylon Translation Software | Privacy Statement | Terms of Use

Fig.1. A gaze plot on the home page of the Babylon website segmented by the extended VIPS algorithm [3]

To conduct scanpath analysis on the web, web pages are typically divided into their areas that can interest or attract users (AOIs) and then fixations are represented with their corresponding AOIs. For example, Figure 1 shows how the home page of the Babylon website is automatically divided into its AOIs with the extended Vision-based Page Segmentation (VIPS) algorithm [3]. If a user looks at the AOIs M, H and H respectively, then his/her path will be represented as MHH. After representing scanpaths in terms of AOIs, they are further analysed for different purposes, especially calculating a similarity score between two scanpaths, computing transition probabilities between AOIs, detecting patterns in a set of scanpaths, or discovering a single representative scanpath of multiple users [4].

Even though there are different algorithms which aim to detect patterns in multiple scanpaths or discover a representative scanpath for a group of scanpaths, their results typically have low similarities to individual scanpaths [4]. Compared to these algorithms, Scanpath Trend Analysis (STA) discovers the most representative path of multiple users on a web page as its result is the most similar to individual scanpaths [5] (see Section 3). The STA algorithm firstly identifies the AOIs that will be included in the resultant path. The first version of this algorithm identifies these AOIs based on all individual scanpaths. Since high variance in individual scanpaths could negatively affect the representativeness of the resultant path, a new parameter called tolerance level was added to the STA algorithm to allow the identification of the AOIs to be included in the resultant path based on a subset of individual scanpaths instead of all of them. The value of the tolerance level parameter can be adjusted both manually and automatically to achieve the most representative path.

The current implementation of the STA algorithm[1] has no visual user interface and its output is only a sequence of alphanumeric characters that represent AOIs (e.g., ABACD). This algorithm is novel in this field and is being increasingly used. For example, it has been used for identifying common patterns while reading programming code [6]. It has also been used to investigate differences in web interaction of people with autism and neurotypical people [7, 8]. Although the STA algorithm is developed for the analysis of eye tracking data, it has also been adapted to analyse sequenced binary motion sensor data to recognise daily activities (meal preparation, eating, dish washing, etc.) of elderly people who live alone [9]. To understand and interpret the output, researchers and practitioners should remember which alphanumeric character corresponds to which AOI. If a visual user interface is developed for the STA algorithm and its output is also visualised on a corresponding page, then we eliminate the burden of remembering and understanding the sequence of letters. Furthermore, some platform and modules are also needed to be installed in advance to run the current implementation.

To address the limitations of the current implementation of the STA algorithm and make it more usable and functional, we presented the first web-based visualisation tool for this algorithm called Visualisation of Scanpath Trend Analysis or shortly ViSTA along with its initial evaluation in our previous paper [10]. It uses the STA algorithm without the tolerance level parameter. This tool does not need any specific platform and modules installed. It allows users to upload an eye tracking dataset and select one of the web pages included in the dataset for further analysis. Since gaze plots are commonly used to visualise scanpaths, this tool visualises the individual scanpaths on the selected page by using gaze plots [2]. It then allows users to visually draw AOIs on the page and apply the STA algorithm for discovering the trending path based on the drawn AOIs. The trending path is then visualised by using an AOI graph. Since an AOI graph shows both AOIs and transitions between these AOIs, it would be easier to recognise which AOIs are trending among users and in which order these AOIs are visited. We evaluated the ViSTA tool by comparing its workload with the workload of the current implementation of the STA algorithm. Our evaluation showed that the STA algorithm can be used with less workload in terms of NASA Task Load Index (TLX) with the ViSTA tool.

In this paper, we present the extended version of the ViSTA tool (See Section 4) along with its follow up evaluation (See Section 5). The extended version of the tool uses the STA algorithm with the tolerance level parameter. In the follow up study, the same methodology of the previous study was used with different users. The results of the follow up study were consistent with the results of the first study which means that the ViSTA tool decreases the workload of using the STA algorithm.

The ViSTA tool can be used by eye tracking researchers, usability evaluators, data scientists and psychologists. Although it is mainly designed for visualising the STA algorithm, it has an open-architecture such that other similar algorithms can also be integrated into this tool. Similarly, other visualisation techniques can also be integrated so that the outputs of the algorithms can be visualised in different ways. We conclude our paper with a discussion of such possible future work (see Sections 6 and 7).

II. Related Work

A large number of techniques have been proposed in the literature to visualise eye tracking data on 2D and 3D visual stimuli in context or not in context [2]. Heat maps and gaze plots have been commonly used in eye tracking research for visualisation. Heat maps use spatial information of eye tracking data, specifically x, y and possibly z coordinates of fixations. These maps use different colours to differentiate which areas are commonly used and which areas are rarely used by users. For example, Tobii Studio Software [11] generates heat maps which show the commonly used areas with the red colour and the rarely used areas with the green colour. These maps can be generated based on different features, such as fixation counts or fixation durations. Although the most commonly used areas can easily be recognised with heat maps, these maps do not take sequential information into consideration. Specifically, heat maps cannot be used to determine in which order these areas are used. Even though gaze plots use both spatial and sequential information

[1] STA's current implementation: https://github.com/SukruEraslan/sta

as illustrated in Figure 1, they will overlap each other when there are many scanpaths. Therefore, the visualisation of many scanpaths with gaze plots at the same time would be a problem because it would be difficult to analyse them. There are also some visualisation techniques which are designed to visualise eye tracking data based on given AOIs [2, 12]. In particular, each AOI can be used as a node (represented with circles) in a graph and directed edges (represented with lines) between these nodes are used to illustrate transitions between AOIs [2]. The sizes of nodes can be determined based on different features, such as fixation count or dwell time [2]. The thickness of edges can also be determined based on different features, such as the number of transitions [13, 2]. This visualisation technique is called an AOI graph [2].

Even though some of these visualisation techniques have a web-based interface, for example [14, 15], most of them do not have a web-based interface which would allow researchers and practitioners to directly access and use them in their studies without installing extra platform and modules on their computers. There are also different algorithms available to process and analyse eye tracking data, especially scanpaths [4]. However, the results of these algorithms are typically not visualised with any visualisation technique on the web platform, with some exceptions [16, 17, 15]. For example, the ScanGraph tool[2] finds similarities between individual scanpaths based on one of three methods (Levenshtein, Needleman-Wunsch and Damerau-Levenshtein) and produces a graph where similar scanpaths are connected to each other [16]. Another example is from [15] which generates an AOI graph based on transitions between AOIs where a ticker edge show more transitions.

To the best of our knowledge, none of the algorithms which discover a representative scanpath for multiple scanpaths has a tool to visualise their outputs on the web. In our previous paper, we presented a web-based visualisation tool for the STA algorithm which is able to identify the most representative path for multiple users, and then compared the workload of this tool with the current implementation of the STA algorithm with a user study [10]. In this paper, we present the extended version of this tool by integrating the tolerance level parameter of the STA algorithm and also present a follow-up user study with more users for workload comparison between the ViSTA tool and the current implementation of the STA algorithm. The STA algorithm is briefly explained in the following section where the full description can be found in [5, 18].

III. STA: Scanpath Trend Analysis

The STA algorithm is a multi-pass algorithm with three main stages which are responsible for preparing individual scanpaths, identifying trending AOIs and constructing the trending path with the trending AOIs respectively [5, 18].

A. Preliminary Stage

The first stage represents each fixation with its corresponding AOI and each individual scanpath is represented a series of AOIs, such as P [250 ms] Q [120 ms] R [300 ms] where P, Q and R are the AOIs fixated by a particular users for 250 ms, 120 ms and 300 ms respectively.

Eye trackers may experience some inaccuracy in storing the exact positions of fixations, and therefore their degree of accuracy are typically provided. Since this situation can cause some problems in identifying the corresponding AOI for each fixation, the STA algorithm takes the degree of accuracy into account and calculates an error offset by using the following formula (Equation 1). It then uses this error offset to extend the AOIs horizontally and vertically. If a fixation falls into more than one AOI because of the extensions, its corresponding AOI will be the closest one to the fixation.

$$E = \tan(x) \cdot D \cdot P \qquad (1)$$

where:

E: Error offset

x: Degree of accuracy

D: Distance between a participant and a screen

P: Pixels per inch for the screen which is calculated based on the screen resolution and the screen size.

B. First Pass

The second stage identifies trending AOIs to be used for the construction of the trending path. An AOI can be visited more than once by a particular user consecutively (such as, P**QQ**R) and/or non-consecutively (such as, P**Q**R**Q**). Each non-consecutive visit to a particular AOI is referred to as an instance. For example, PQRRPQP contains three instances of P, two instances of Q and one instance of R. These instances are numbered to be differentiated based on the dwell time (i.e., total fixation duration on each instance) where the first number is given to the longest instance of a particular AOI (such as, P1 Q2 R1 R1 P2 Q1 P3).

The first version of the STA algorithm considers all individual scanpaths for identifying trending AOI instances. Specifically, if a particular AOI instance gets at least the same attention as the fully shared AOI instances in terms of the total fixation duration and the total number of fixations, it is considered as trending AOI. However, the second version of the STA algorithm does not have to identify trending instances based on fully shared instances. In the second version of the STA algorithm, the following conditions should be satisfied by an instance to be trending.

- The total number of fixations on the instance is greater than or equal to the minimum total number of fixations on the instances which are shared by a subset (determined based on the tolerance level parameter) of the individual scanpaths.
- The dwell time on the instance is greater than or equal to the minimum dwell time on the instances which are shared by a subset (determined based on the tolerance level parameter) of the individual scanpaths.

[2] ScanGraph: www.eyetracking.upol.cz/scangraph

The value of the tolerance level parameter can be between zero and one. When the tolerance level is set to 0.5, the algorithm identifies trending instances based on the instances which are shared by 50% of individual scanpaths. The algorithm does not only allow to adjust this parameter manually, but also adjusts it automatically to identify the most representative trending path (i.e, the most similar path to individual scanpaths).

The instances which are not trending among users are removed from the individual scanpaths as they will not be used for constructing the trending path.

C. *Second Pass*

The last stage constructs the trending scanpath by combining the trending AOI instances based on their overall positions in the individual scanpaths. The same instances in each scanpath are firstly merged (such as, P1 [250 ms] P1 [150 ms] Q1 [200 ms] R1 [270 ms] → P1 [400 ms] Q1 [200 ms] R1 [270 ms]), and the sequential priority value is computed for each instance with the following formula (Equation 2):

$$\psi_i = 1 - P_i \cdot z, where\ z = \frac{max_i - min_i}{L - 1} \qquad (2)$$

where:

ψ_i: Priority value of ith instance

P_i: Position of the instance in the scanpath (start: 0)

L: Length of the scanpath

max_i: Maximum priority value (default: 1)

min_i: Minimum priority value (default: 0.1)

Once the individual priority values of the instances are computed in each of the scanpaths, the total priority value (Ψ) for each instance is computed with the following formula (Equation 3):

$$\Psi = \sum_{i=1}^{n} \psi_i \qquad (3)$$

where:

Ψ : Total priority value

n: The number of scanpaths

The trending path is then constructed by combining the trending instances and sorting them according to their total priority values in descending order. The dwell time and the total number of fixations on the instances are also considered in case of the same total priority value. In the end, the instance numbers are removed (such as, P1 → P) and the consecutive repetitions are deleted (such as, PQQR → PQR) such that the trending path is represented in terms of the AOIs.

IV. VISTA: VISUALISATION OF SCANPATH TREND ANALYSIS

The ViSTA tool[3] can be used to upload an eye tracking dataset, visualise individual scanpaths with gaze plots, visually draw AOIs on stimuli, apply the STA algorithm to discover the trending scanpath and then visualise the trending path with an AOI graph. As illustrated in Figure 2, the ViSTA tool is comprised of three main modules explained below which are as follows: (1) Data Processing, (2) Algorithmic Processing and (3) Visualisation Processing. This tool was mainly developed by using jQuery which is a JavaScript library. The current implementation of the STA algorithm was also slightly modified by using Flask[4] to run it as a server which accepts POST requests from the ViSTA tool.

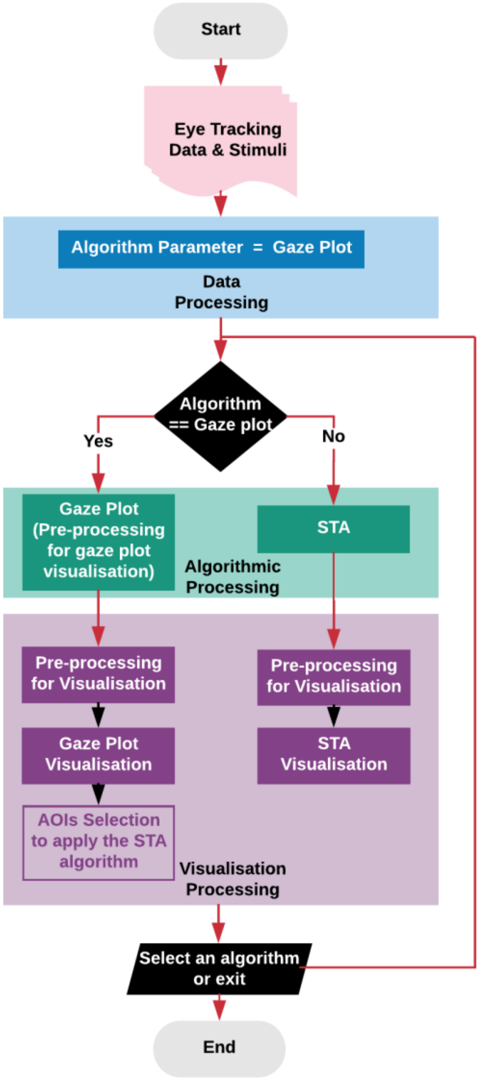

Start
Eye Tracking Data & Stimuli
Algorithm Parameter = Gaze Plot
Data Processing
Algorithm == Gaze plot
Yes
No
Gaze Plot (Pre-processing for gaze plot visualisation)
STA
Algorithmic Processing
Pre-processing for Visualisation
Pre-processing for Visualisation
Gaze Plot Visualisation
STA Visualisation
AOIs Selection to apply the STA algorithm
Visualisation Processing
Select an algorithm or exit
End

Fig.2. The work flow of the ViSTA tool [Selection of AOIs is needed when the STA algorithm will be applied.]

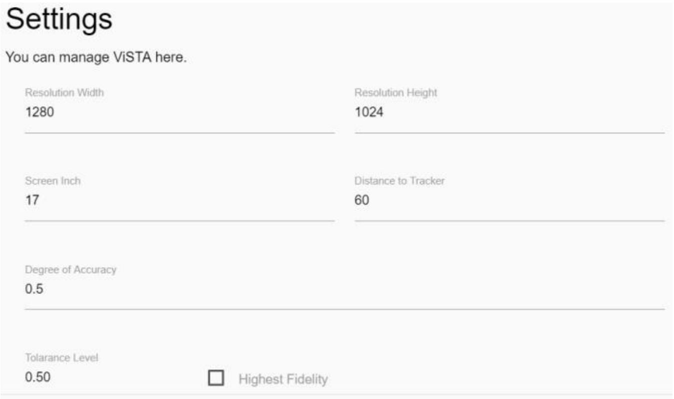
A. *Data Processing*

The first module is responsible from taking eye tracking data and converting this data into an internal storage format for further analysis. There should be a different data file for

[3] ViSTA: http://iam.ncc.metu.edu.tr/vistatool/demo/index.html
[4] Micro web framework for Python: http://flask.pocoo.org/

each participant that contains the details of their fixations (index, timestamp, duration, x and y coordinates and stimuli name). These data files should include web page links as stimuli name since the ViSTA tool is currently designed for scanpath analysis on the web. This tool accesses each page by using their links to retrieve some information, especially their titles. When the data files are uploaded and they are converted into an internal storage format, this module becomes ready to send the data to the next module for further processing.

In this tool, two algorithms are currently available: (i) the gaze plot algorithm which further processes the data to visualise individual scanpaths with gaze plots; and (ii) the STA algorithm which further processes the data to identify the trending scanpath. Since the ViSTA tool firstly visualises individual scanpaths with gaze plots, the algorithm parameter is set to the gaze plot algorithm internally here so that individual scanpaths are visualised with gaze plots in the first round. The ViSTA tool also allows setting the parameters of the STA algorithm as shown in Figure 3 (see the parameters of the STA algorithm in Section 3). Specifically, when the "Highest Fidelity" option is chosen, the STA algorithm automatically adjusts the tolerance level to identify the most representative path.



Settings

You can manage ViSTA here.

Resolution Width
1280

Resolution Height
1024

Screen Inch
17

Distance to Tracker
60

Degree of Accuracy
0.5

Tolarance Level
0.50

□ Highest Fidelity

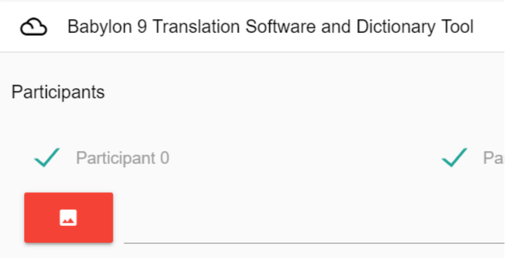
Fig.3. The settings page of the ViSTA tool to set the parameters of the STA algorithm

B. *Algorithmic Processing*

When the Algorithmic Processing module takes the data, it further processes the data based on the given algorithm parameter. If the algorithm parameter is set to the gaze plot algorithm, this module further processes the data to be able to visualise individual scanpaths with gaze plots. However, if the algorithm parameter is set to the STA algorithm, this module further process the data to discover the trending path. Therefore, the algorithm processing modules either prepares the data to be visualised with gaze plots or applies the STA algorithm and prepares the result to be visualised with an AOI graph based on the algorithm parameter.

C. *Visualisation Processing*

The Visualisation Processing module takes the results of the algorithms and visualises them with their visualisation techniques by using vis.js[5]. In particular, when the results of the gaze plot algorithm are received, researchers and practitioners should firstly select which participants' scanpaths on which web pages will be visualised with gaze plots. By default, all the participants are selected. When the participants and the web page are selected, this tool takes a screen shot of the page with html2canvas[6] by using its link and the individual scanpaths are visualised on the page with gaze plots. In case of any problem with taking a screen shot of the page, the tool allows to upload an image file for the page as shown in Figure 4 (the red button). This is also an improvement to the first version of the ViSTA tool that was presented in [10]. Figure 5 shows how the ViSTA tool visualises individual scanpaths on the home page of the Babylon website with gaze plots. Researchers and practitioners can then visually draw AOIs to be used with the STA algorithm. If no AOI is drawn, the STA algorithm cannot be selected for further processing the data because this algorithm discovers the trending scanpath in terms of the AOIs. As an example, Figure 5 also shows three AOIs identified by a user on the home page of the Babylon website.



Babylon 9 Translation Software and Dictionary Tool

Participants

✓ Participant 0 ✓ Pa

Fig.4. An option to upload an image file for a web page

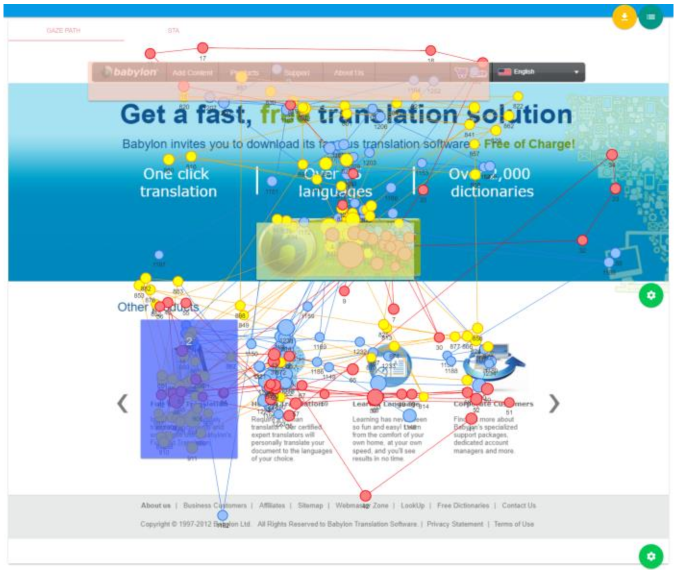
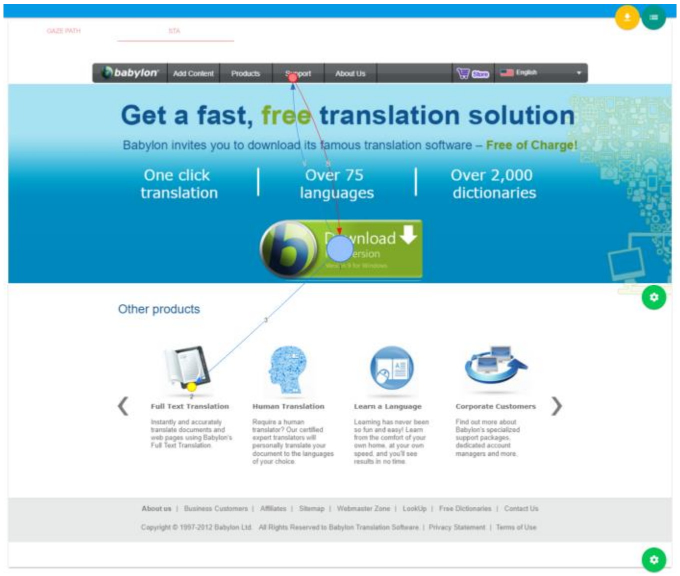

GAZE PATH
STA
babylon
Get a fast, free translation solution
Babylon invites you to download its famous translation software Free of Charge!
One click translation
Over 75 languages
Over 1,000 dictionaries
Other Products

Fig.5. Identification of AOIs on the home page of the Babylon website with gaze plots

When the STA algorithm is selected after the visualisation of individual scanpaths with gaze plots and the result of the STA algorithm is received, this module firstly pre-processes

[5] Dynamic, browser based visualisation library: http://visjs.org/
[6] Screenshots with JavaScript: https://html2canvas.hertzen.com/

the result for visualisation, and then visualises it by using an AOI graph. An example of this visualisation is shown in Figure 6. Since the AOI that includes the link to download the free version of Babylon is the most frequent AOI in the trending path, its node is larger than other two AOIs' nodes.



Get a fast, free translation solution

Babylon invites you to download its famous translation software – Free of Charge!

One click translation | Over 75 languages | Over 2,000 dictionaries

Other products

Full Text Translation: Instantly and accurately translate documents and web pages using Babylon's Full Text Translation.

Human Translation: Require a human translator? Our certified expert translators will personally translate your document to the languages of your choice.

Learn a Language: Learning has never been so fun and easy! Learn from the comfort of your own home, at your own speed, and you'll see results in no time.

Corporate Customers: Find out more about Babylon's specialized support packages, dedicated account managers and more.

About us | Business Customers | Affiliates | Sitemap | Webmaster Zone | LookUp | Free Dictionaries | Contact Us

Copyright © 1997-2012 Babylon Ltd. All Rights Reserved to Babylon Translation Software | Privacy Statement | Terms of Use

Fig.6. Visualisation of the trending scanpath discover by the STA algorithm on the home page of the Babylon website

V. Evaluation

In order to assess the effectiveness of the VISTA tool, two user studies were conducted by using the NASA Task Load Index (TLX). User studies are a reliable way to understand how products are perceived by users [19] and NASA TLX has widely been used to perform workload assessments for different kinds of human-machine systems [20]. This metric is composed of six attributes: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort and Frustration.

In this study, our main research question was "Does the ViSTA tool have better TLX attributes in comparison with the current implementation of the STA algorithm?". We investigated this research question with the following questions associated with each of the TLX attributes:

- Mental Demand: "How mentally demanding was the task?"
- Physical Demand: "How physically demanding was the task?"
- Temporal Demand: "How hurried or rushed was the pace of the tasks?"
- Performance: "How successful were you in accomplishing what you were asked to do?"
- Effort: "How hard did you have to work to accomplish your level of performance?"
- Frustration: "How insecure, discouraged, irritated, stressed, and annoyed were you?"

Each attribute is assessed by the participants by giving a value between one and 20. Higher values represent negative scores. For example, for mental demand one means very low and 20 means very high mental load. This is also the case for the performance attribute - a higher score means the users did not think they were successful in accomplishing the tasks they were asked to do. In the given scale, one means perfect performance and 20 means failure in terms of performance.

We conducted the first study with 12 participants from the Middle East Technical University Northern Cyprus Campus, and presented the results in [10]. To extend the evaluation of the ViSTA tool, we conducted a follow-up study by using the same methodology but with different 12 participants from various universities in Turkey. We present the results of the follow-up study below along with the comparison with the results of the first study. We also combine the results from the first and second studies to present an overall evaluation of the tool. We were able to do this because both studies followed the same methodology.

A. *Methodology*

A detailed description of the dataset used in our user studies for the evaluation of the ViSTA tool can be found in [5]. In both of these user studies, we used a subset of that dataset. First of all, we randomly divided the dataset into two halves called dataset A and dataset B. We then asked our participants to apply the current implementation of the STA algorithm with one half of the dataset and the ViSTA tool with the other half to identify their trending paths on a particular web page. A half of the participants used the current implementation of the algorithm and then the ViSTA tool whereas another half used the ViSTA tool and then the current implementation of the algorithm. Besides this, a half of the participants applied the ViSTA tool with the dataset A and the current implementation of the algorithm with the dataset B whereas another half applied the ViSTA tool with the dataset B and the current implementation of the algorithm with the dataset A. Therefore, we ensured that all the cases were counterbalanced to deal with any possible familiarity and order effects. When the participants identified the trending scanpath for a particular dataset, they were given a printed copy of the page to draw the trending path on the page and then they were asked to fill in the English version of the NASA TLX Form[7].

We also applied to the dependent T-Test or its non-parametric alternative Wilcoxon signed-rank test to statistically compare the workload of the ViSTA tool and the current implementation of the STA algorithm based on NASA TLX attributes. Since we expected to have lower workload with the ViSTA tool, we applied these tests as one-sided. The statistical comparison was not applied in our previous paper [10], so it is also a new analysis for the first user study.

1) *Participants*

The first study was conducted at Middle East Technical University Northern Cyprus Campus with four female and eight male students. The participants were between the age of 18-24, apart from one of them who was between the age of 25-

[7] https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLXScale.pdf

34. All of the participants were daily web users. They were also asked to assess their computer skills by using a 5-point Likert scale. The mean value was 3.42 with the standard deviation 1.00.

The second study was conducted with three female and nine male students from different universities in Turkey. Nine of these participants were between the age of 18-24 whereas three of them were between the age of 25-35. All of them were also daily web users. They ranked their computer skills as 3.17 out of 5 (standard deviation: 0.72)

2) Procedure

The participants firstly read the information sheet to understand the main objectives of the study and their rights and then they signed a consent form. After that, they were asked to complete a short questionnaire to collect their basic demographic information (gender, age groups, web usage, general computer skills, and departments). A training session was then given to them to illustrate how they can use the current implementation of the STA algorithm and the ViSTA tool. This training session mainly used a sample dataset (not the one used in the evaluation). After the training session, they started their evaluation session.

3) Materials

The original eye tracking dataset includes six web pages with varying level of visual complexities. We used the home page of the Babylon website for the training sessions, and the home page of the Apple website for the evaluation sessions.

B. RESULTS

This section presents the results from the first and the second studies respectively and then combines the results of these studies to provide an overall evaluation of the ViSTA tool.

1) Results of Evaluation Study I

Figure 7 and Table I shows the results from the first study. Figure 7 shows the comparison of the current implementation of the STA algorithm and the ViSTA tool based on the mean values of the NASA TLX attributes where the error bars illustrate the standard deviation. Besides this, Table I shows a detailed descriptive analysis of the NASA TLX attributes for the current implementation of the STA algorithm and the ViSTA tool.

The results of the first study shows that the ViSTA tool needs less mental demand, physical demand and temporal demand in comparison with the current implementation of the STA algorithm. The participants also stated that they needed less effort to complete the given task with the ViSTA tool. Moreover, they were less frustrated when they used the ViSTA tool. However, they stated that they performed slightly better (better satisfied with their performance in achieving the task) with the current implementation of the STA algorithm.

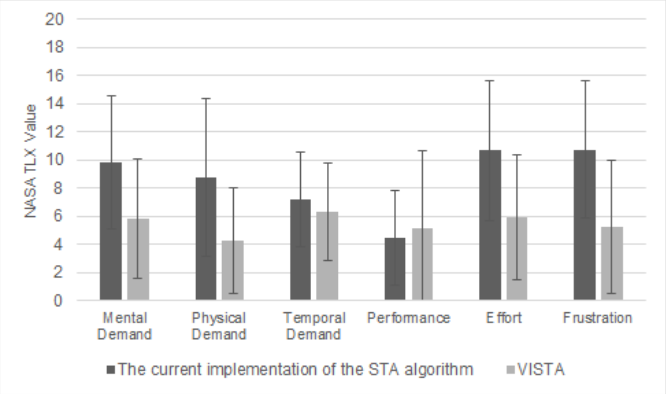


Fig.7. NASA TLX attributes for the current implementation of the STA algorithm and the ViSTA tool from the first study

TABLE I
THE MEAN (M), MEDIAN (MD) & STANDARD DEVIATION (SD) OF THE NASA TLX ATTRIBUTES IN THE FIRST STUDY

	The current imp. of the STA alg.			**ViSTA**		
NASA TLX Attributes	**M**	**MD**	**SD**	**M**	**MD**	**SD**
Mental Demand	9.8	9.5	4.7	5.8	4.5	4.2
Physical Demand	8.8	10.0	5.6	4.3	3.0	3.8
Temporal Demand	7.2	8.0	3.4	6.3	5.5	3.5
Performance	4.5	3.5	3.4	5.2	2.5	5.5
Effort	10.7	12.5	5.0	5.9	4.5	4.4
Frustration	10.8	10.5	4.9	5.3	3.5	4.7

The statistical comparison shows that the VISTA tool has a significantly lower workload in terms of mental demand (t(11)=-2.65, p<0.05, d=0.8), physical demand (t(11)=-2.8, p<0.05, d=0.84), effort (t(11)=-3.06, p<0.05, d=0.92) and frustration level (t(11)=-3.45 p<0.05, d=1.04) with large effect size compared to the current implementation of the STA algorithm. Although the temporal demand is also lower with the ViSTA tool, it is not statistically different (t(11)=-1.22, p=0.12, d=0.37). Besides, there is no significant difference between the ViSTA tool and the current implementation of the STA algorithm in terms of their performance (t(11)=0.55, p=0.3, d=0.16).

2) Results of Evaluation Study 2

Figure 8 and Table II illustrates the results from the second study. Figure 8 illustrates the comparison of the current implementation of the STA algorithm and the VISTA tool in based on the mean values of the NASA TLX attributes where the error bars illustrate the standard deviation. Moreover, Table II illustrates a detailed descriptive analysis of the NASA TLX attributes for the current implementation of the STA algorithm and the ViSTA tool.

Similar to the results of the first study, the results of the second study also illustrates that the participants completed their tasks with less mental demand, physical demand, temporal demand, effort and frustration with the ViSTA tool in comparison with the current implementation of the STA algorithm. However, in contrast to the first study's participants, the second study's participants stated that they performed slightly better with the ViSTA tool instead of the current implementation of the STA algorithm.

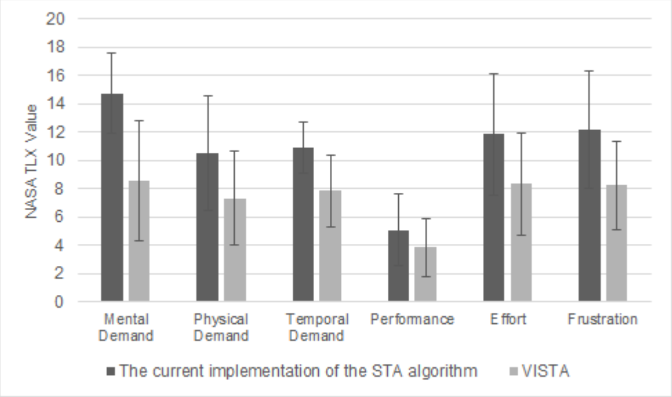


Fig.8. NASA TLX attributes for the current implementation of the STA algorithm and the ViSTA tool from the second study

TABLE II
THE MEAN (M), MEDIAN (MD) & STANDARD DEVIATION (SD) OF THE NASA TLX ATTRIBUTES IN THE SECOND STUDY

	The current imp. of the STA alg.			ViSTA		
NASA TLX Attributes	**M**	**MD**	**SD**	**M**	**MD**	**SD**
Mental Demand	14.8	16.0	2.8	8.6	7.5	4.2
Physical Demand	10.5	12.0	4.0	7.3	7.5	3.3
Temporal Demand	10.9	11.0	1.8	7.8	7.0	2.6
Performance	5.1	5.0	2.6	3.8	3.0	2.0
Effort	11.8	14.0	4.3	8.3	9.0	3.6
Frustration	12.2	13.0	4.2	8.3	6.5	3.1

The statistical comparison shows that the VISTA tool has significantly better values for mental demand (t(11)=-4.21, p<0.05, d=1.27), physical demand (t(11)=-2.7, p<0.05, d=0.81), temporal demand (t(11)=-3.99, p<0.05, d=1.2), effort (t(11)=-4.31, p<0.05, d=1.3) with large effect size and frustration (z= 2.09, p<0.05, r=0.43) with medium effect size in comparison with the current implementation of the STA algorithm. Although the ViSTA tool has a better performance value at this time, it is not statistically different (t(11)=-1.26 p=0.12, d=0.38).

3) Overall Results

Figure 9 and Table III shows the overall results by combining the results of both the first and second studies. The overall results suggest that the ViSTA tool scores significantly better in mental demand (t(23)=-4.83, p<0.05, d=1.01), physical demand (t(23)=-3.9, p<0.05, d=0.81), effort (t(23)=-4.76, p<0.05, d=0.99) and frustration (t(23)=-4.5, p<0.05, d=0.94) with large effect size and temporal demand (t(23)=-3.52, p<0.05, d=0.73) with medium effect size in comparison with the current implementation of the STA algorithm. Even though the ViSTA tool compared to the current implementation of the STA algorithm scores slightly better in performance in the second study, its performance score is slightly lower in the first study. However, when we combine the results of both of the studies to provide an overall evaluation, the ViSTA tool performance was perceived better (i.e., the overall performance score was lower for ViSTA which means better performance), even though the difference is not statistically significant (t(23)=-0.37 p=0.36, d=0.08).

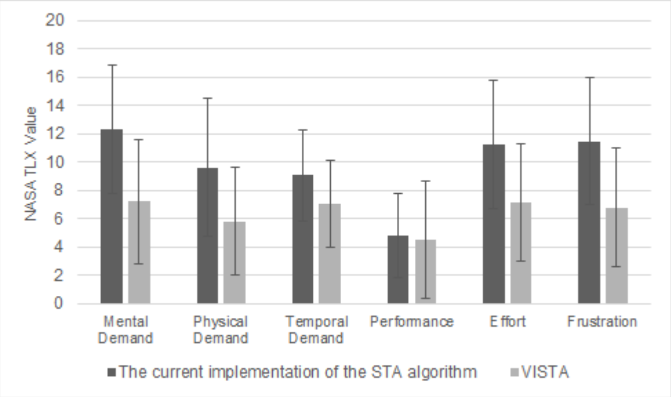


Fig.9. Overall NASA TLX attributes for the current implementation of the STA algorithm and the ViSTA tool from both the first and second studies

TABLE II
THE OVERALL MEAN (M), MEDIAN (MD) & STANDARD DEVIATION (SD) OF THE NASA TLX ATTRIBUTES

	The current imp. of the STA alg.			ViSTA		
NASA TLX Attributes	**M**	**MD**	**SD**	**M**	**MD**	**SD**
Mental Demand	12.3	13.0	4.5	7.2	6.0	4.4
Physical Demand	9.6	11.0	4.8	5.8	5.0	3.8
Temporal Demand	9.0	10.0	3.3	7.1	6.5	3.1
Performance	4.8	4.0	2.9	4.5	3.0	4.1
Effort	11.3	13.0	4.6	7.1	7.5	4.1
Frustration	11.5	12.0	4.5	6.8	6.0	4.2

VI. Discussion

The ViSTA tool provides a web-based interface for the STA algorithm. It allows researchers and practitioners to directly access and use the STA algorithm for their studies without installing extra platform and modules on their computers. Similar to other algorithms, the STA algorithm can also be used by different researchers in different fields. Therefore, it should not be expected that all of these researchers easily manage to download and run their implementations on their computers. With the ViSTA tool, researchers and practitioners can easily upload their eye tracking datasets and visually draw their AOIs to apply the STA algorithm. When the trending path is generated with the algorithm, it is visualised with an AOI graph so that researchers can easily interpret the trending path without the need for remembering which alphanumeric character represents which AOI as they will see the trending AOIs and the order in which the AOIs are used. Our overall evaluation results show that the ViSTA tool decreases the workload of using the STA algorithm.

The current version of the ViSTA does not include automatic AOI detection based on fixation clusters or the source code of web pages. The tool allows researchers to visually draw their own AOIs on the fly. They may detect AOIs by using different techniques and then they can draw the AOIs in the ViSTA tool. As the tool has an open-architecture, it can be easily extended in the future to support automatic AOI detection. For example, the VIPS algorithm has been extended and implemented as a web service [3], so we can use this web service to automatically segment web pages into their areas based on their source code and visual representation. Appropriate clustering algorithms can also be implemented as part of this tool to allow the identification of AOIs by clustering fixations [21]. Another visualisation level can also be added to the ViSTA tool for AOIs which will be shared by other components of the tool.

The ViSTA tool currently has only the STA algorithm, apart from the generic algorithm to pre-process the data for visualising individual scanpaths with gaze plots. However, it has an open architecture which means that other algorithms can easily be integrated. For example, the SPAM algorithm has been used to identify sequential patterns in eye tracking data [22]. It can also be integrated into this tool and its output can also be visualised with an AOI graph. However, some studies need to be conducted. Similarly, if there are other online web-based eye tracking tools, the ViSTA tool can also easily be integrated to those tools. It is a web application so it can easily be added to other web-based applications.

We are still improving this tool by exploring appropriate algorithms and visualisation techniques to make this tool more functional and usable. In the future, we are planning to add different visualisation techniques to the Visualisation Module (see Figure 2), such as a time plot [23]. As a consequence, when a particular algorithm provides an output, a list of appropriate visualisation techniques can be listed to be selected for visualising the output. The visualisation of the outputs of the algorithms can also be animated to make them more informative. For example, when the output of the STA algorithm is visualised with an AOI graph, its nodes and edges can be shown one at a time based on the time sequence which allows researchers to focus on a particular trending AOI at a time during analysis. The ViSTA tool can also be improved in a way that a particular algorithm can be applied to different groups of participants (at the moment, the STA algorithm is only applied to one group) and the results of these groups can be visualised on the same visual stimulus for a comparison purpose. Additionally, an account system can be created such that the users of the tool can have their workbenches. This would allow them to store their datasets and revisit their workbenches.

Our work presented here is not without limitations. We evaluated the ViSTA tool with 24 people in total who were mostly not very experienced in eye tracking research and studies. This situation could have an impact on the scores given by our participants. However, we believe if we had eye-tracking specialists, our results would be even better for the ViSTA tool. However, to confirm this, in the future, we are planning to conduct further studies with more, and more experienced participants for detailed feedback.

VII. Conclusion

The aim of this paper is to introduce the extended version of the first web-based interface of the STA algorithm and also present its extended user evaluation where the participants were asked to assess the workload of the VISTA tool and the current implementation of the STA algorithm for the comparison purpose. The STA algorithm is novel in this field and it is being increasingly used in different studies. Its visualisation, called ViSTA, will allow researchers and practitioners to use the STA algorithm with a lower workload and understand its output directly without the need of remembering the names of AOIs and relating the AOIs to the output in their minds. Since there is a limited number of web-based tools for analysing eye tracking data, our tool also makes an useful contribution to eye tracking research. This tool is open to be further improved by integrating other algorithms and visualisation techniques.

References

[1] Y. Yesilada, S. Harper and S. Eraslan, “Experiential Transcoding: An EyeTracking Approach,” in *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, New York, NY, USA, 2013.

[2] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf and T. Ertl, “Visualization of Eye Tracking Data: A Taxonomy and Survey,” *Computer Graphics Forum*, vol. 36, pp. 260-284, 2017.

[3] M. E. Akpınar and Y. Yeşilada, “Vision Based Page Segmentation Algorithm: Extended and Perceived Success,” in *Current Trends in Web Engineering: ICWE 2013 International Workshops ComposableWeb, QWE, MDWE, DMSSW, EMotions, CSE, SSN, and PhD Symposium, Aalborg, Denmark, July 8-12, 2013. Revised Selected Papers*, Q. Z. Sheng and J. Kjeldskov, Eds., Cham, Springer International Publishing, 2013, pp. 238-252.

[4] S. Eraslan, Y. Yesilada and S. Harper, “Eye tracking scanpath analysis techniques on web pages: A survey, evaluation and comparison,” *Journal of Eye Movement Research,* vol. 9, 2015.

[5] S. Eraslan, Y. Yesilada and S. Harper, “Scanpath Trend Analysis on Web Pages: Clustering Eye Tracking Scanpaths,” *ACM Trans. Web,* vol. 10, pp. 20:1--20:35, 11 2016.

[6] C. Tablatin and M. M. Rodrigo, “Identifying Common Code Reading Patterns using Scanpath Trend Analysis with a Tolerance,” in *Proceedings of thee 26th International Conference for Computers in Education (ICCE 2018)*, Metro Manila, Philippines, 2018.

[7] S. Eraslan, V. Yaneva, Y. Yesilada and S. Harper, “Do Web Users with Autism Experience Barriers When Searching for Information Within Web Pages?,” in *Proceedings of the 14th Web for All Conference on The Future of Accessible Work*, New York, NY, USA, 2017.

[8] S. Eraslan, V. Yaneva, Y. Yesilada and S. Harper, “Web users with autism: eye tracking evidence for differences,” *Behaviour & Information Technology,* vol. 38, pp. 678-700, 2019.

[9] H. Y. Yatbaz, S. Eraslan, Y. Yesilada and E. Ever, “Activity Recognition Using Binary Sensors for Elderly People Living Alone: Scanpath Trend Analysis Approach,” *IEEE Sensors Journal,* 2019.

[10] Ş. Eraslan, S. Karabulut, M. C. Atalay and Y. Yeşilada, “ViSTA: Visualisation of Scanpath Trend Analysis (STA) / Scanpath Trend Analysis (STA)'in Görselleştirilmesi,” in *Proceedings of the 12th Turkish National Symposium on Software Engineering (12. Ulusal Yazılım Mühendisligi Sempozyumu, UYMS 2018)*, İstanbul, Turkey, 2018.

[11] Tobii Technology AB, “Tobii Studio™ 2.X User Manual (Sep. 2010),” 2010.

[12] M. Burch, A. Kull and D. Weiskopf, “AOI rivers for visualizing dynamic eye gaze frequencies,” in *Computer Graphics Forum*, 2013.

[13] K. Holmqvist, J. Holsanova, M. Barthelson and D. Lundqvist, “Reading or scanning? A study of newspaper and net paper reading.,” J. R. In Hyönä and H. Deubel, Eds., Elsevier, 2003, pp. 657-670.

[14] L. Herman, S. Popelka and V. Hejlova, “Eye-tracking Analysis of Interactive 3D Geovisualization,” *Journal of Eye Movement Research,* vol. 10, 2017.

[15] M. Burch, A. Kumar and N. Timmermans, “An Interactive Web-based Visual Analytics Tool for Detecting Strategic Eye Movement Patterns,” in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications,* New York, NY, USA, 2019.

[16] J. Dolezalova and S. Popelka, “ScanGraph: A Novel Scanpath Comparison Method Using Visualisation of Graph Cliques,” *Journal of Eye Movement Research,* vol. 9, 2016.

[17] G. Topić, A. Yamaya, A. Aizawa and P. Martínez-Gómez, “FixFix: Fixing the Fixations,” in *Proceedings of the Ninth Biennial ACM Symposium on ETRA*, New York, NY, USA, 2016.

[18] S. Eraslan, Y. Yesilada and S. Harper, “Engineering web-based interactive systems: trend analysis in eye tracking scanpaths with a tolerance,” in *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2017,* New York, NY, USA, 2017.

[19] S. Eraslan and C. Bailey, “End-User Evaluations,” in *Web Accessibility: A Foundation for Research*, Y. Yesilada and S. Harper, Eds., London, : Springer London, 2019, pp. 185-210.

[20] A. Cao, K. K. Chintamani, A. K. Pandya and R. D. Ellis, “NASA TLX: Software for assessing subjective mental workload,” *Behavior Research Methods,* vol. 41, pp. 113-117, 01 2 2009.

[21] A. Santella and D. DeCarlo, “Robust Clustering of Eye Movement Recordings for Quantification of Visual Interest,” in *Proceedings of the 2004 Symposium on ETRA,* New York, NY, USA, 2004.

[22] P. Hejmady and N. H. Narayanan, “Visual Attention Patterns During Program Debugging with an IDE,” in *Proceedings of the 2012 Symposium on ETRA*, New York, NY, USA, 2012.

[23] K.-J. Räihä, A. Aula, P. Majaranta, H. Rantala and K. Koivunen, “Static Visualization of Temporal Eye-Tracking Data,” in *Human-Computer Interaction - INTERACT 2005: IFIP TC13 International Conference, Rome, Italy, September 12-16, 2005. Proceedings*, M. F. Costabile and F. Paternò, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 946-949.

BIOGRAPHIES

Şükrü Eraslan is currently working as a research associate at the University of Manchester. He completed his BSc in Information Systems Engineering at Cyprus International University in 2009. After that, he received his MSc in Information Technology (Data Engineering) at the University of Wolverhampton in 2011. He then completed his Ph.D. in Computer Science at the University of Manchester in 2016. After his Ph.D., he worked as a post-doctoral researcher at Middle East Technical University Northern Cyprus Campus. His main research area is eye tracking on the Web. He is interested in analysing eye movements of users to understand how they interact with web pages to make the web more accessible in constrained environments. Besides, he has started to work on learning analytics & software engineering.

Serkan Karabulut received his B.Sc. in Computer Engineering from Middle East Technical University Northern Cyprus Campus (METU NCC) in 2012. He is currently working in AYESAŞ as a Software Developer. His research interests include eye tracking and human-computer interaction.

Mehmet Can Atalay received his B.Sc. in Computer Engineering from Middle East Technical University Northern Cyprus Campus (METU NCC) in 2012. He is currently doing his M.Sc. in Cognitive Science at Middle East Technical University (METU) and working as a Software Engineer in Turkey. His research interests include deep learning, neuro-linguistic programming and eye tracking.

Yeliz Yeşilada is an Associate Professor at Middle East Technical University Northern Cyprus Campus (METU NCC) and an honorary research fellow in the School of Computer Science at the University of Manchester. She received her B.Sc. in Computer Engineering from the Eastern Mediterranean University in 1999, her M.Sc. in Computer Science from the University of Manchester in 2000 and also completed her Ph.D. in 2005 in Computer Science at the University of Manchester. Her primary research interest is centred around Human Computer Interaction and User Experience; in particular the mobile Web, Web accessibility and eye tracking research to improve user experience. Further information about Dr. Yeşilada can be found at: http://www.yelizyesilada.info.