EyeClouds: A Visualization and Analysis Tool for Exploring Eye Movement Data

Conference Paper · September 2019

DOI: 10.1145/3356422.3356423

3 authors:

Michael Burch
University of Applied Sciences of the Grisons FHGR (Fachhochschule Graubünden)

Alberto Veneri
Università Ca' Foscari Venezia

Bangjie Sun
National University of Singapore

EyeClouds: A Visualization and Analysis Tool for Exploring Eye Movement Data

Michael Burch, Alberto Veneri, and Bangjie Sun
Eindhoven University of Technology
Eindhoven
m.burch@tue.nl

ABSTRACT

In this paper, we discuss and evaluate the advantages and disadvantages of several techniques to visualize and analyze eye movement data tracked and recorded from public transport map viewers in a formerly conducted eye tracking experiment. Such techniques include heat maps and gaze stripes. To overcome the disadvantages and improve the effectiveness of those techniques, we present a viable solution that makes use of existing techniques such as heat maps and gaze stripes, as well as attention clouds which are inspired by the general concept of word clouds. We also develop a web application with interactive attention clouds, named the EyeCloud, to put theory into practice. The main objective of this paper is to help public transport map designers and producers gain feedback and insights on how the current design of the map can be further improved, by leveraging on the visualization tool. In addition, this visualization tool, the EyeCloud, can be easily extended to many other purposes with various types of data. It could be possibly applied to entertainment industries, for instance, to track the attention of the film audiences in order to improve the advertisements.

CCS CONCEPTS

- **Human-centered computing** → **Visualization techniques**;

KEYWORDS

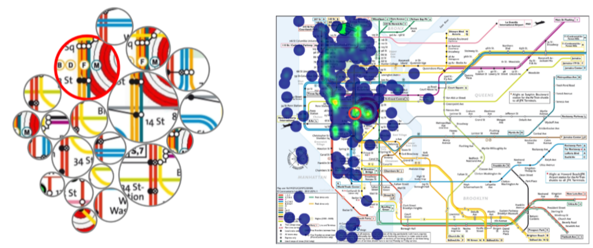
Visualization, Attention cloud, Heat map, Gaze stripes, Public transport map, Eye movement data

ACM Reference Format:
Michael Burch, Alberto Veneri, and Bangjie Sun . 2019. EyeClouds: A Visualization and Analysis Tool for Exploring Eye Movement Data. In *VINCI '19: The 12th International Symposium on Visual Information Communication and Interaction, September 20–22, 2019, Shanghai, China,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3231622.3231630

VINCI '19, September 20–22, 2019, Shanghai, China

ACM ISBN 978-1-4503-6501-7/18/08. . . $15.00
https://doi.org/10.1145/3231622.3231630

1 INTRODUCTION



(a) (b)

Figure 1: The attention cloud and heat map of the New York City transport map: (a) shows the new visualization proposed in this paper, i.e. an attention cloud of the eye movement data, and (b) shows the corresponding visual attention map.

Analyzing people's reaction to public transport maps can be a challenging and tedious task for map designers and producers [13, 21], given the growing amount of spatio-temporal eye movement data [15, 17]. It is difficult to identify common eye movement behavior among a group of map viewers [6, 8, 9] because eye movement data usually consists of several dimensions, such as a stimulus, the position of a fixation point, and the duration.

It is also troublesome for data analysts to constantly refer back to the map in order to understand the meaning of a fixation record. Furthermore, data analysts may look at the data from different perspectives, and hence, there is no one-size-fits-all approach. To support such an identification process of common eye movement behavior, an interactive visualization tool can be a good strategy.

In this paper, we present our viable solution to visualize eye movement data recorded in a public transport map eye tracking experiment. In the following sections, we will discuss the design of the visualization techniques used, data processing and handling, and the architecture and implementation details of the interactive web application, the EyeCloud (see Figure 1).

In general, the web application consists of three major features for visualization: an attention cloud [14], a visual attention map [3], and gaze stripes [20]. With the term attention cloud we refer to a visual representation of the data very similar to the approach used in a tag cloud or a word cloud but created with, instead of words, thumbnails of an image. The size of each thumbnail is dependent on the visual attention strength of the user to the corresponding section of the map. This new visualization technique gives data analysts a new way to rapidly retrieve information about eye movement data.

While the attention cloud gives the analysts a rapid and intuitive overview of the most gazed points on the map, the visual attention map helps them to quickly understand the context of those points as well as the overall distribution of the gazed points and the relative visual attention strength. The gaze stripes, on the other hand, provide the analysts with additional information on the scanpaths of map viewers, as well as the relationship between time stamps and fixation points. The EyeCloud combines these three visualization techniques supporting the data analysts to quickly and easily identify common eye movement behaviors among viewers.

2 RELATED WORK

There are many visualization techniques for eye movement data [1] but not all of them focus on finding a common pattern among a group of people like in the work by Burch et al. [5, 7, 11, 12]. But actually, finding common visual scanning strategies [19] as well as outliers [10] is a good way to analyze eye movement data.

One of the common techniques is the visual attention map or heat map [3, 4]. It is useful in finding visual attention hot spots as it shows the spatial distribution of the visual attention against the locations on the map. However, one of the negative effects is that it is not possible for analysts to compare different map viewers over time. If those were drawn for each map viewer individually, it would be difficult and tedious for the analysts to compare all different heat maps and find out the common pattern of visual attention. Another disadvantage of heat maps is that the actual content of the map is hidden from the user, and as a result, it is troublesome for the analysts to constantly refer back to the original map just to relate the visual attention strength to the actual content of the map.

Another technique comes with the gaze stripes that represent eye movement data [20] by a series of thumbnails of a map based on the gazed points and time stamps. Since a gaze stripe contains time information and is designed for only one map viewer, the most obvious disadvantage of that is the scalability. Given a large data set with a huge amount of map viewers, and for each viewer, a large number of fixation points, the size of the gaze stripes could be too large to deal with [22]. One issue is that the scanpaths might not be readable anymore if we fit them into a smaller viewport, because each of the thumbnails might only occupy a few pixels or even smaller, and hence, the information of a thumbnail is lost. Another issue is that it might be very tedious for the analysts to trace the gaze stripes because if we want to maintain the information of each thumbnail, the size of the gaze stripes would be extremely large, and common patterns could not be easily seen.

In this paper, we propose a method to visualize common patterns in the eye movement data by comparing the visual attention strength at different locations of a map and provide an interactive linking to associate such strength with the actual map content and extra visual representations like visual attention maps [3] and gaze stripes [20]. This new approach is based on the well-known tag cloud visualization technique [14] that has been used in the past and will probably be used in the future for a multitude of purposes, e.g. for visualizing the semantic field of a resource in social media applications and for trend visualization [18].

3 DESIGN AND ARCHITECTURE

The EyeCloud web application consists of several components to visualize and analyze eye movement data by exploring the common visual attention pattern among study participants. To reach our objective, we have to compare eye movement data, visualize and order the comparison, and finally allow interactions and linkings to the original public transport map.

3.1 Design Criteria

To achieve above-mentioned objectives, five requirements have been formulated to access whether the visualization tool is necessary and appropriate in meeting our goals.

- **Selection of maps and viewers:** The user is able to select the public transport map and the viewers.
- **Overview and comparison of scanpaths:** The user is able to have an overview of the scanpaths from selected participants and compare them.
- **Interactivity and responsiveness:** The application is interactive to allow the user to have as much control as possible, and highly responsive to minimize loading time of the application and waiting time of the user.
- **Clear and relevant output:** The visualization output is clear and relevant to give insights to the user.
- **Extend ability and scalability:** The application is extendable and scalable for different use cases and data sets.

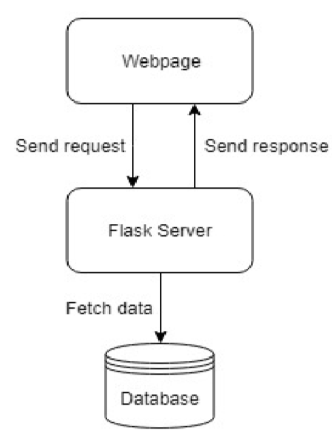


Figure 2: A high-level diagram of the EyeCloud application

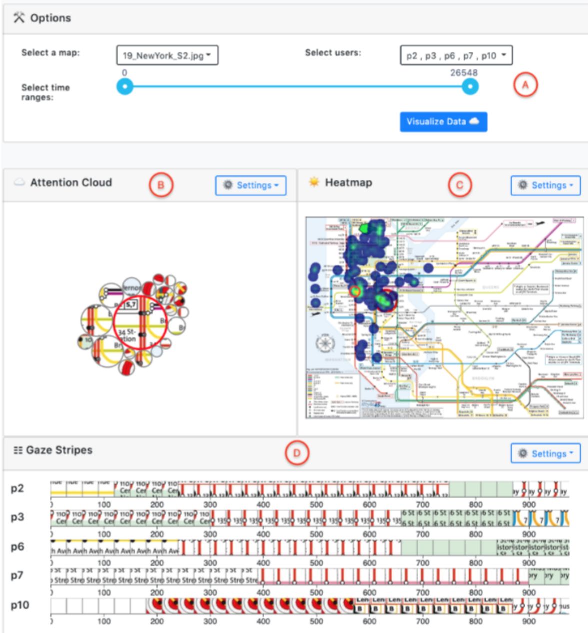


Figure 3: Front-end design with labeled components

3.2 Reproducibility

All the source code is available at the following repository: https://github.com/veneres/EyeCloud with MIT License. In the root folder of the repository, there is all the documentation needed to run the server and the client inside the file README.md. To make our implementation reproducible we have provided a useful Docker file that is one of the latest de facto tools to provide the product that we have made according to the so-called 'DevOps' philosophy.

3.3 Architecture

The application is developed as a web application with the Single Page Application (SPA) approach, to avoid interruption of the user experience between successive pages and to behave like a desktop application. As interaction with the SPA often involves dynamic communication with the web server behind the scenes, the back-end is needed to provide RESTful web services for the front-end side. Figure 2 describes that the front-end web-page (client) will communicate with the backend (server) by sending a request and getting a response, and the back-end will fetch data from the database and process data retrieved. Such an architecture provides the calibration of the workload and the possibility to be scalable without overloading the client side with resource-intensive operations that could be performed out of the stage.

3.3.1 Back-End. The back-end consists of two parts: a database to store eye movement data and a Flask server (http://flask.pocoo.org/). The database has been developed with MongoDB, a cross-platform document-oriented database management system that uses JSON-like documents with schemata.

The advantage of using MongoDB instead of a common relational DBMS is that it is highly capable of being integrated with a front-end application written in JavaScript, easy to maintain and highly scalable with the addition of new features and increasing amounts of data [15]. The Flask server, written in Python, processes data upon requests and responds with a set of JSON documents to the front-end. A cache mechanism has been implemented to improve the performance of data processing and to reduce wastage of computation during the heat map generation.

3.3.2 Front-End. The front-end has been developed in TypeScript using Angular with the help of the D3.js library (https://d3js.org/). Using a well-defined framework to develop the application is useful to create a structured application with the possibility to implement future features without rethinking the entire application and, at the same time, adopting one of the most popular JavaScript libraries for data visualization. The front-end of our application mainly consists of four parts: (A) a control panel, (B) a heat map, (C) an attention cloud, and (D) gaze stripes.

The layout of the main parts is shown in Figure 3. Every part has been developed as an Angular component, integrated by three Angular directives and two Angular services.

4 DATA MODEL AND HANDLING

In this section, we explain the data that we have used to create a working example for our application. The following data has been used for the development of this project:

- A collection of stimuli images in .JPG format
- A text file collecting the information about the complexities of images
- A CSV file with the fixation data

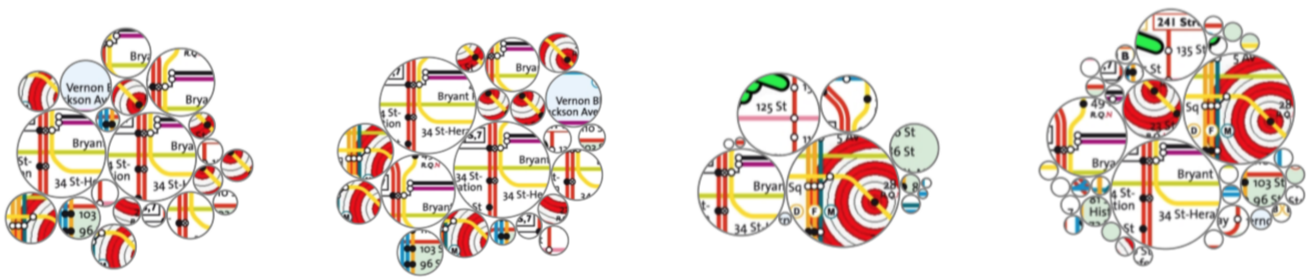
Other possible improvements could be done if other metadata would be available for each user such as the age, the country of origin, and other possible meaningful data in order to create a larger number of filters for the end user, i.e. the designer, however, the visualization tool created contains enough options that can be parameterized with respect to our purposes.

4.1 Data Preprocessing

In this project we have assumed that the data passed to our web application is consistent and free of incorrectness, e.g. we assumed that the coordinates provided with the fixation data are valid and the x- and y-coordinates of each point are inside the resolution of the corresponding stimulus. This assumption is given by the fact that the data that we have used inside the application was already preprocessed, therefore no further manipulation was needed.

4.2 Data Integration and Transformation

The data has been transformed and integrated inside the application and stored inside the database using a Python



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

Figure 4: Attention clouds of the New York transport map displayed with different options: (a) Default display. (b) Different cropping size. (c) Different cluster radius. (d) Different number of maximum points displayed.

script and the main two collections created, i.e. the fixation and station collections.

In our database, we have created other two collections: a collection to represent the users and a collection to store the precomputed heat maps. However, due to the fact that we do not have any specific information about the users, the first is a simple collection with a document with only one primary key and, regarding the second one, it is a simple dictionary representation of the pixel matrix of the heat maps.

The 'Settings' panel allows users to select: map, map viewers, and time range. The attention cloud, heat map, and gaze stripes will be re-created from the filtered data.

4.3 Data Aggregation

From the perspective of data aggregation needed to create the heat map and the gaze stripe we have simply referred to existing algorithms described by Kurzhals et al. [20] for the gaze stripes and by Blignaut [2] for the heat map.

However, naturally, there is no existing aggregation algorithm implemented for the new visualization proposed in this paper and the aggregation method could considerably affect the outcome offered to the user. The implemented algorithm is inspired by the fact that the cluster radius is a parameter of our visualization so we just create the fixation point cluster with this constructive constraint.

5 VISUALIZATION TECHNIQUES

The main functionality of our web application is to visualize eye movement data with an attention cloud, although this is not sufficient to provide to the end user other types of visualizations. The users should be able to select a subset of data for the visualization on demand and all visual diagrams should be correctly generated within a short period of time or, at least, a loading screen has to be shown to the user to indicate the loading of the new visualization.

5.1 Attention Cloud

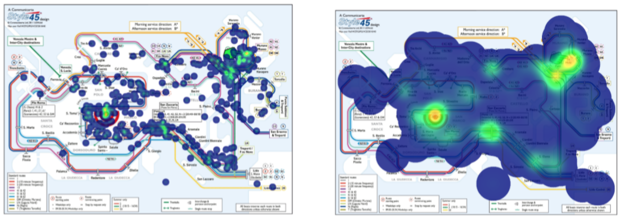
The attention cloud will display images of different cropping sizes with larger ones representing the parts of the map where people stare at for the longest time. One image will contain the area around a fixation point, and the size of the area will be proportional to the fixation duration of a viewer on that point. As a result, the most common area of the map will appear the most obvious in the attention cloud. The generation of the attention cloud makes use of the force simulation function from the D3 library that allows us to draw force-directed graphs [16].

In our case, we represent the attention cloud as a graph with nodes represented by thumbnails and without edges. For each thumbnail, we decided to make the shape to be circular because the visual span of fixation used in the heat map is circular, and forms visual correlations between the two components. The variable parameters used in the procedure are fixPoints, maxCrop, minCrop, and maxThumb.

First of all, we aggregate the fixation points by taking the maximum duration from the newly created variable aggreFixPoints and we start to iterate on each cluster (that are sorted per fixation duration). For each cluster, we decide the corresponding thumbnail size and we start to positioning each thumbnail anticlockwise creating a circumference made with k elements. Defining N as the size of aggrFixPoints we have a series of points on the circumference made of k thumbnails each on which we simply apply the force-directed graph drawing algorithm [16].

The application allows the user to change various visualization options: Max cropping size, min cropping size, cluster radius, and number of data points.

In Figure 4 we show four examples of different visualization option combinations. In (a) the attention cloud has been created with the default option, i.e. maxCrop = 100, minCrop = 20, clusterRadius = 0, and maxThumb = 20. In (b) in order to make the fixation points bigger, they are built with a higher duration, from the default configuration we increase the maxCrop to 120 and we decrease minCrop to 10. In (c) with the purpose of having thumbnails that are slightly different, e.g. the two biggest points in the middle of (a), we adjust the clusterRadius to 100 pixels. Finally, in (d) we changed the number of points displayed to 50 and we reduce the cluster radius to 40 with the aim of seeing how many points are outside our clusters.



(a) (b)

Figure 5: Different heat map settings for the visual span: (a) Heat map with visual span at 20px. (b) Heat map with visual span at 100px.

5.2 Heat Map

A heat map displayed at the right side of the attention cloud shows the distributions of people's attention to various spots on the map, with color closest to red representing the most attention paid to the spot while green representing the least. The heat map helps users to quickly identify the 'hottest' locations on the map and their corresponding images in the attention cloud.

The generation of the pixel mask of the heat map has been implemented following the guidelines given by visual span and other parameters and passed to the front-end to be immediately rendered in a Canvas HTML object. The creation of the pixel mask requires a lot of computational resources because for every fixation point we have to compute the weight of the nearest pixels in a fixed radius (defined as a parameter of the function) according to the probability that an observer will perceive certain pixels for the given fixation, i.e. the farther the coordinates of the pixel will be from the center of the fixation point the smaller will be the probability that the user will perceive this pixel.

Thus, the computational complexity of this calculation is O(d^2n) where d is the diameter of the circumference that defines the fixation point and n is the number of the fixation points. To overcome this computational problem we have cached those results inside the MongoDB database as a matrix summary in a JSON format, however, no further shrewdnesses have been taken into consideration since the optimization of the heat map generation is out of the scope of our project and during the loading a simple loading animation is displayed.

Two options are available: Visual span radius in pixels and the possibility to hide or show the heat map (see Figure 5). The visual span radius integration has been implemented as described by Blignaut [2] and we also allow the end user to view the actual content of the map.

5.3 Gaze Stripes

The gaze stripes, consisting of a series of thumbnails, will form the scan paths of people who see the map, giving insights into what people pay more attention to within a selected period of time. We have added this view to give the possibility to analyze also the individual scan path of each user that would be lost in the spatio-temporal aggregation in the heat map and in the attention cloud (see Figure 6).

(a)

(b)

Figure 6: Different gaze stripes: (a) Gaze stripes with scale set at 10px. (b) Gaze stripes with scale set at 40px.

Our implementation is based on the one described by Kurzhals et al. [20] with one novel feature. In our work every gaze stripe is individually draggable. We have implemented this feature to facilitate the comparison between two gaze stripes from different users in different periods of time. This feature could be very useful to get insights when analyzing the different periods of time without visual occlusion because, for example, we allow the analyst to compare the first 3 seconds of fixation points from a user and the last 3 seconds from another user to find recurring patterns.

In this visualization there are two options available:

- **Scale:** the thumbnail dimension in terms of pixels
- **Granularity:** the duration of each thumbnail

The scale proposed is from 10 to 100 pixels for each thumbnail and, setting m as the minimum duration from the selected fixation points, from $\frac{1}{5}$ m to m. In addition, it is also possible to display the gaze stripes with different x-axes (the axes to represent the time) setting the granularity option to 0, thus each gaze stripe will be displayed with an axis scaled according to the minimum duration of its thumbnail.

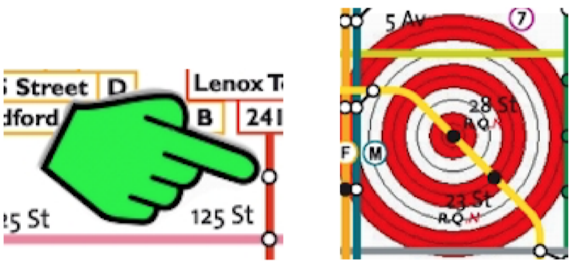
5.4 Interaction Techniques

In addition to the interactions available for each component already exhaustively explained, below we list the possible interactions that users can make with the EyeCloud web application:

- **Attention cloud interaction:** Clicking the image in the attention cloud will show the corresponding point on the heat map.
- **Heat map interaction:** Clicking a point on the heat map will highlight the corresponding image in the attention cloud.
- **Gaze stripe interaction:** Clicking on a thumbnail on the gaze stripe will highlight the corresponding point on the heat map and on the attention cloud.

- **Data selection:** Changing settings such as map, map viewers, or time range will force to re-create the attention cloud, heat map, and gaze stripes.

When users click on the image in the attention cloud, the web application will make use of the front-end cache to search for the corresponding point in the heat map and change the styling of the point to make it distinct from the others. The same process can be applied to heat map interaction and the gaze stripes interaction. Other interactions could be interesting to develop such as the thumbnail exclusion from the attention cloud, i.e. the possibility of removing a thumbnail from the attention cloud and its consequent recreation, but the interactions developed are already sufficient to give the analyst a large number of parameters to retrieve some useful insights.



(a) (b)

Figure 7: Example of representation of the start and ending point on the maps: (a) Start point for New York map. (b) End point for New York map.

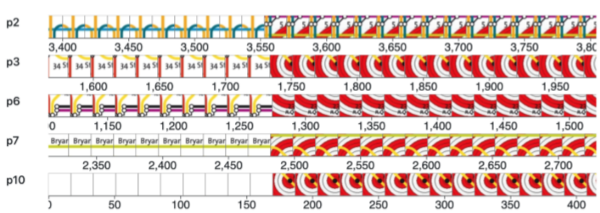


Figure 8: Gaze stripes of 5 different users for the New York transport map

6 APPLICATION EXAMPLES

In this section, we will explore real-world eye movement data sets, and visualize them with our application to retrieve information and gain insights. We have chosen three different representative maps with several complexity levels (high, medium, low).

6.1 New York (High Complexity)

New York City (NYC) is the most populous and modern city in the United States with an estimated population of over 8.6 million, distributed over 784 square kilometers. It is also the most densely populated major city in the United States. This fact is well illustrated by the high complexity of the public transport map of NYC, and indeed the complexity is the highest in our data set. In addition, NYC received an eighth consecutive annual record of 62.8 million tourists in 2017.

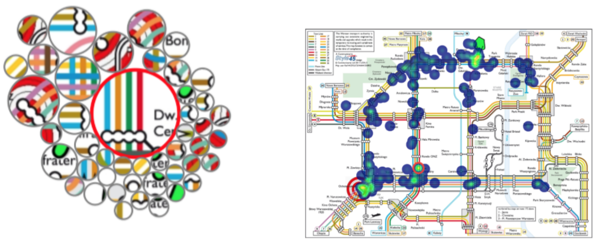
As we all know, a public transport map is a necessity for any tourist to the city, and a user-friendly map with the good design would certainly be a benefit for most tourists. The question to map designers and producers is what and how to improve the design of the map. Take a close look at the EyeCloud and see what information we can get from the three visual components. We are interested in finding out common eye movement patterns among different map viewers if they want to travel from one place to another. Take note that the start point of the journey is represented as a green hand and the destination is illustrated as a red aiming target, as shown in Figure 7.

Firstly, we can get an overall pattern of eye movement from the heat map shown in Figure 3. We can see that most points are clustered at the left side, forming a clear path from the start point to the end point, with a small number of outliers (fixation points that are not on the path). It suggests that map viewers are able to find the desired path and keep on track most of the times, indicating that the overall design of the map is good. Then, let's take a closer look at the attention cloud in Figure 3.

The most obvious thumbnails are the bigger ones gathered at the center of the cloud. They represent the sections of the map that viewers spend a relatively long time looking at. In this example, we can see that the most commonly gazed points among a group of selected map viewers are showing an interchange station, followed by the endpoint, and the start point is rarely looked at. It is the greatest advantage of an attention cloud that the data analysts are able to identify the most commonly gazed points in just one glance. Moreover, we can also compare the sizes of those thumbnails and realize that they are much larger than the rest.

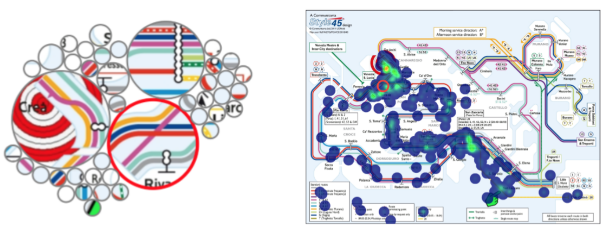
After clicking on the biggest thumbnail in the attention cloud, the corresponding location is also highlighted on the heat map, and it is indeed the hottest spot. We can also click on other thumbnails to compare the attention cloud and the heat map to gain more information. From the two visual components, the general message that we receive is that the map viewers spend a lot of time staring at the interchange station and relatively short time on the end point. Hence, it suggests that it is relatively challenging for map viewers to figure out the path via the interchange station given the current design of the representation of interchange stations on the map, and it is the part that can be further improved.

However, some analysts may not be satisfied and want to get more details on scanpaths of each map viewer with respect to time stamps. They can then take a look at the gaze stripes. Since the gaze stripes are individually draggable, it allows the analysts to easily compare eye movement patterns at different time stamps. In Figure 8 we see that after dragging and aligning gaze stripes of the five selected map viewers, we can observe that all viewers have looked at the end point, but at very different time stamps. For instance, participant 10 (p10) looks at the destination at the very beginning, while the rest looks at the destination at much later time. It suggests that a common behavior among those selected viewers is that they tend to look at the destination near the end and it may be because they want to confirm that the path they choose is able to reach the destination.



(a) (b)

Figure 9: Attention cloud (a) and heat map (b) of Warsaw, the biggest thumbnail in the attention cloud is an intersection point



(a) (b)

Figure 10: Attention cloud (a) and heat map (b) of Venice, two of the three biggest points in the attention cloud are intersection points

6.2 Warsaw (Medium Complexity)

Warsaw is the capital and largest city of Poland. Its population is officially estimated at 1.8 million residents within a greater metropolitan area of 3.1 million residents, which makes Warsaw the eighth-most populous capital city in the European Union. Warsaw is an alpha global city, a major international tourist destination, and a significant cultural, political, and economic hub. That is why the Warsaw transport map is chosen as the representative of maps with medium complexity.

Similar to how we analyze the New York City map, we can get a big picture of eye movement behavior by looking at the heat map first. From the heat map shown in Figure 9, fixation points of the map viewers form two possible paths on the left half of the map from the start point to the end point, but there is also a relatively large number of fixation points scattered around the right half of the map. Those points seem to form another two possible paths.

Such an observation is interesting because the destination is on the left side of the map and it should be natural for a viewer to search for paths on the left which are supposed to be shorter than other possibilities. However, the heat map shows that map viewers also look at the two possible paths on the right, and it suggests how confused the map viewers can be, given the current design of the map.

Furthermore, if we look closely at the actual map content of the fixation points on the heat map, most of them are actually interchange stations. To get a quick insight, we can switch our view to the attention cloud. Indeed, the biggest thumbnail at the center of the cloud indicates an interchange station. We can also notice that most big thumbnails show interchange stations and multiple transport lines crossing each other at different locations, while only a few of them are showing the start and end points. This suggests that map viewers spend too much time on the intersections of lines, trying to figure out which way to go. Since we know from the heat map that map viewers are confused with multiple choices of the possible paths, the attention cloud then shows that the main problem can be the design of transport lines and interchange stations that might be unclear and non-reader-friendly to map viewers.

To gain more insights, we can compare the New York City map and Warsaw map. It is noticeable that map viewers spend a relatively long time on interchange stations for both maps. However, it is surprising that, even with lower complexity levels, map viewers of the Warsaw map are actually more confused in choosing their way from the start point to the destination than those of the New York City map. The most obvious problem of the Warsaw map is the unclear transport lines that are crossing each other and hence, it is very hard to trace. Another problem is that every station looks like an interchange station because multiple transport lines are drawn together. If we look at the New York City map, we can clearly distinguish between a normal station and an interchange station.

6.3 Venice (Low Complexity)

Venice is one of the biggest cities in the northern part of Italy with an approximate population of 260,000 inhabitants considering the whole metropolitan area, approximately one-sixth of the population of Warsaw. However, there are 60,000 tourists per day and an estimate of 25 million every year. We have chosen Venice because even if it has a low complexity level the actual transportation system in the city is very complex. The interesting fact about the map of Venice is that public transport is not provided with a standard bus or tram but with water bus (also called water taxi). Starting the analysis of the heat map we can easily see that people find two main paths to go from the starting point to the end point: one is through the external part of the island and the other is within the city. Most of the people chose the latter path since maybe it looks like the shortest one on the map.

Moreover, every legend has some fixation points on it and one of the legends has also a relatively strong visual attention. This fact suggests that the colors used to represent the transportation lines are not self-explanatory and some map viewers have to refer to the legends to understand the meaning of the different colors. Thus, it may imply that the map needs a new design to make the paths more understandable. Let's take a look at the attention cloud to get an overview of the fixation points and we can notice in a glance that there are three relatively big thumbnails (see Figure 10): the ending point and two interchange stations.

Comparing the attention cloud with that of New York and Warsaw we can find that the biggest thumbnail for Venice is the ending point but it is not the case for the other two. This is probably due to the fact that the ending point in Venice is occluded by some transportation lines that make map viewers difficulties to understand if it is the right destination. In addition, similar to the previous two examples, the other two biggest thumbnails in the attention cloud are interchange stations indicating that, for the cases studied, one of the greatest challenges is to understand how to proceed from an interchange station.

7 CONCLUSION AND FUTURE WORK

We have created a web application, the EyeCloud, in which the user can select a public transport map and multiple map viewers to gain an overview of common visual attention patterns and distributions of visual attention strength. This can be linked to locations on the map, to compare scanpaths between participants, while the results are depicted in an attention cloud, a heat map, and gaze stripes. Those visual components can be reconstructed by selecting a different subset of maps and viewers. The objective was to create a tool to help public transport map designers and producers to visualize and analyze eye movement data to gain insights in how the design of the maps can be improved. The user has full control over the parameters, giving freedom to find patterns and information in ways we might not even expect. There are several ways that could improve this application. Firstly, it is great to give the analysts more options to extract information from any eye movement data set. Possible options are additional descriptive statistics of the maps and viewers because sometimes numbers tell more than pictures. It is also possible to allow the analysts to compare two maps against each other and to select subsets of map viewers and paths within both. Furthermore, this visualization tool can be easily extended to many other purposes with various types of data.

REFERENCES

[1] Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum* 36, 8 (2017), 260–284.

[2] Pieter J. Blignaut. 2010. Visual span and other parameters for the generation of heatmaps. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA*. 125–128.

[3] Agnieszka Bojko. 2009. Informative or Misleading? Heatmaps Deconstructed. In *Human-Computer Interaction – INTERACT*. Springer, 30–39.

[4] Michael Burch. 2016. Time-Preserving Visual Attention Maps. In *Proceedings of Conference on Intelligent Decision Technologies*. 273–283.

[5] Michael Burch. 2017. Mining and visualizing eye movement data. In *Proceedings of SIGGRAPH ASIA Symposium on Visualization*. 3:1–3:8.

[6] Michael Burch. 2017. Visual Analysis of Eye Movement Data with Fixation Distance Plots. In *Proceedings of Conference on Intelligent Decision Technologies*. 227–236.

[7] Michael Burch. 2018. Identifying Similar Eye Movement Patterns with t-SNE. In *Proceedings of Vision, Modeling & Visualization, VMV*. 111–118.

[8] Michael Burch, Gennady L. Andrienko, Natalia V. Andrienko, Markus Höferlin, Michael Raschke, and Daniel Weiskopf. 2013. Visual Task Solution Strategies in Tree Diagrams. In *Proceedings of IEEE Pacific Visualization Symposium*. 169–176.

[9] Michael Burch, Andreas Kull, and Daniel Weiskopf. 2013. AOI Rivers for Visualizing Dynamic Eye Gaze Frequencies. *Computer Graphics Forum* 32, 3 (2013), 281–290.

[10] Michael Burch, Ayush Kumar, Klaus Mueller, Titus Kervezee, Wouter Nuijten, Rens Oostenbach, Lucas Peeters, and Gijs Smit. 2019. Finding the outliers in scanpath data. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA*. 83:1–83:5.

[11] Michael Burch, Ayush Kumar, and Neil Timmermans. 2019. 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, ETRA*. 93:1–93:5.

[12] Michael Burch, Kuno Kurzhals, Niklas Kleinhans, and Daniel Weiskopf. 2018. EyeMSA: exploring eye movement data with pairwise and multiple sequence alignment. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA*. 52:1–52:5.

[13] Michael Burch, Kuno Kurzhals, and Daniel Weiskopf. 2014. Visual Task Solution Strategies in Public Transport Maps. In *Proceedings of ET4S@GISCIENCE*. 32–36.

[14] Michael Burch, Steffen Lohmann, Daniel Pompe, and Daniel Weiskopf. 2013. Prefix Tag Clouds. In *Proceedings of International Conference on Information Visualisation, IV*. 45–50.

[15] Andrew T. Duchowski. 2003. *Eye Tracking Methodology - Theory and Practice*. Springer.

[16] Thomas M. J. Fruchterman and Edward M. Reingold. 1991. Graph Drawing by Force-directed Placement. *Software - Practice and Experience* 21, 11 (1991), 1129–1164.

[17] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost van de Weijer. 2011. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press.

[18] Shah Khusro, Fouzia Jabeen, and Aisha Khan. 2018. Tag Clouds: Past, Present and Future. In *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*. 1–13.

[19] Ayush Kumar, Neil Timmermans, Michael Burch, and Klaus Mueller. 2019. Clustered eye movement similarity matrices. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA*. 82:1–82:9.

[20] Kuno Kurzhals, Marcel Hlawatsch, Florian Heimerl, Michael Burch, Thomas Ertl, and Daniel Weiskopf. 2016. Gaze Stripes: Image-Based Visualization of Eye Tracking Data. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 1005–1014.

[21] Rudolf Netzel, Bettina Ohlhausen, Kuno Kurzhals, Robin Woods, Michael Burch, and Daniel Weiskopf. 2017. User Performance and Reading Strategies for Metro Maps: An Eye Tracking Study. *Spatial Cognition & Computation* 17, 1–2 (2017), 39–64.

[22] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. 2005. Feature Congestion: A Measure of Display Clutter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 761–770.