REGULAR PAPER



Michael Burch · Alberto Veneri · Bangjie Sun

Exploring eye movement data with image-based clustering

Received: 27 February 2020/Revised: 27 April 2020/Accepted: 8 May 2020/Published online: 5 July 2020 © The Author(s) 2020

Abstract In this article, we describe a new feature for exploring eye movement data based on image-based clustering. To reach this goal, visual attention is taken into account to compute a list of thumbnail images from the presented stimulus. These thumbnails carry information about visual scanning strategies, but showing them just in a space-filling and unordered fashion does not support the detection of patterns over space, time, or study participants. In this article, we present an enhancement of the EyeCloud approach that is based on standard word cloud layouts adapted to image thumbnails by exploiting image information to cluster and group the thumbnails that are visually attended. To also indicate the temporal sequence of the thumbnails, we add color-coded links and further visual features to dig deeper in the visual attention data. The usefulness of the technique is illustrated by applying it to eye movement data from a formerly conducted eye tracking experiment investigating route finding tasks in public transport maps. Finally, we discuss limitations and scalability issues of the approach.

Keywords Visualization \cdot Attention cloud \cdot Image-based clustering \cdot Public transport map \cdot Eye movement data

1 Introduction

Exploring eye movement data (Duchowski 2003; Holmqvist et al. 2011) is pretty challenging due to its spatio-temporal nature (Blascheck et al. 2015). However, finding insights in the strategic behavior of eye tracked people (Burch 2017b; Burch et al. 2013a, b) can bring various benefits, for example, finding design flaws in a stimulus that hinder people to rapidly process the visual information or to interpret it.

To design and build a useful approach to this data exploration problem, we come up with a list of research questions:

- **RQ1: Focus on data dimensions space, time, and participants** Is it possible to build a tool that provides visual perspectives on the typical data dimensions in eye movement data?
- **RQ2: Linking of the views on the data dimensions** Can a linking between those visualizations be beneficial to detect strategic eye movement patterns for an individual person or a group of eye tracked people?
- **RQ3: More structured views on the extracted data** Would a more structured representation like a clustering based on image thumbnails provide even more benefits for the data exploration?

• **RQ4: Easy-to-understand approach** Can additional simple statistical diagrams as well as standard easy-to-understand visual variables (Munzner 2014; Tufte 1992) help to make the approach useful for non-experts in visualization?

Interactive visualizations of eye movement data (Blascheck et al. 2017) can support data analysts to detect normal or abnormal patterns, but each visualization is just designed for a certain task at hand (Ware 2008). In particular, detecting patterns over space, time, and the study participants as well as providing detailed and scalable information about the presented stimulus is a difficult task. Algorithmic preprocesses of the data and visual encodings of the results can interplay to provide the best possible ways to interpret the data, to build, confirm, reject, or refine hypotheses (Keim 2012), and finally, derive knowledge from the recorded eye movement data.

This article presents an extended version of the EyeCloud technique (Burch et al. 2019c) by adding, as major ideas among others, an image-based clustering concept as well as the temporal information to the original word cloud-based visualizations. The detailed extensions can be summarized as follows:

- **Image-based clustering** The visually attended stimuli regions are extracted and the list of image thumbnails is further algorithmically processed and clustered based on image properties like average pixel RGB values.
- **Temporal thumbnail links** To guarantee a view on the temporal aspects in the image thumbnails, we added interactive color-coded links that indicate the sequential and chronological order among the thumbnails.
- Attention bar chart Each thumbnail is attached by a visual attention frequency which is visually encoded as a bar chart. Those bars can be used as filter function to modify the views for showing more relevant stimulus regions.
- Focus mode To not distract the viewer with unneeded image information of the stimulus, we can only display the visually attended regions and gray out or remove the rest. This helps to move the focus to the important stimulus aspects.
- Extra statistical information Further values about visual attention can be derived and shown as a details-on-demand view. This extra information can lead the observer to further insights and give more textual output.
- Further interaction techniques Apart from the already long list of interaction techniques, we integrated even more. For example, temporal filtering, different clusterings, as well as typical interactions in the novel visual concepts are now included.

We describe the extended technique (see Fig. 1), provide details about the clustering approach, apply it to eye movement data from a formerly conducted eye tracking study investigating the readability of public transport maps (Burch et al. 2014; Netzel et al. 2017), and discuss scalability issues and limitations. We will also discuss the design of the visualization techniques, data processing and handling, and the architecture and implementation details of the interactive web application, starting with the original EyeCloud approach which is based on word cloud layouts applied to image thumbnails.



Fig. 1 The public transport map of Antwerp is used as a stimulus, all users are selected. 500 thumbnails are generated based on the fixation points with minimum and maximum crop sizes of 20 and 70, respectively. The cluster radius is 0. \mathbf{a} 1 cluster and links of width 4. \mathbf{b} 5 clusters and links of width 4. \mathbf{c} 5 clusters and no links



Fig. 2 A visually attended stimulus can be shown as an EyeCloud consisting of image thumbnails (a). To understand the context of the thumbnails, an interactively linked visual attention map is displayed (b)

In general, the web application consists of three major features for visualization (see examples for an EyeCloud and a visual attention map linked to that in Fig. 2): an attention cloud (Burch et al. 2013c), a visual attention map (Bojko 2009), and gaze stripes (Kurzhals et al. 2016b). The attention cloud builds an image-based overview of the visually attended stimulus regions, but in the EyeCloud technique, the focus was on space-efficiency and space-filling properties of the layout. In the novel idea, we also focus on image-based clustering, bringing similar thumbnails spatially together.

While the attention cloud provides an overview about the visually attended points in the stimulus, the visual attention map helps 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. Finally, 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 (Blascheck et al. 2017) but not all of them focus on finding a common pattern among a group of people like in the work by Burch (2017a, 2018) and Burch et al. (2018b, 2019b). But actually, finding common visual scanning strategies (Kumar et al. 2019) as well as outliers (Burch et al. 2019a) is a good way to analyze eye movement data.

One of the common techniques is the visual attention map or heat map (Bojko 2009; Burch 2016). It is useful for finding visual attention hot spots as it shows the spatial distribution of the visual attention in the locations in a stimulus shown in an eye tracking study. However, one of the negative effects is that it is not possible for analysts to compare different map viewers over time due to aggregations and overplotting. 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.

This aggregation problem is not occurring in gaze plots (Goldberg and Helfman 2010), but as a negative consequence of overplotting all scanpaths on top of the stimulus, we get an effect denoted by visual clutter (Rosenholtz et al. 2005). Various scanpaths cannot be taken into account simultaneously to find design flaws in a given stimulus. Only interaction techniques can help to filter for the most important patterns (Yi et al. 2007) over space, time, and participants or a more aggregated scanpath visualization must be chosen, for example, showing a hierarchical flow (Burch et al. 2018a) or allowing to inspect the eye movement data as an aggregated graph visualization (Burch et al. 2019b). However, the stimulus context information is typically not directly integrated in such a visualization which is one goal of the EyeCloud approach.

Another technique comes with the gaze stripes that represent eye movement data (Kurzhals et al. 2016b) by a series of thumbnails of a stimulus that is based on the gazed points and time stamps. Since a gaze stripe contains time information and is designed for several viewers in parallel, the most obvious disadvantage

comes with visual scalability. Given a large data set with a huge amount of viewers, and for each viewer, a large number of fixation points, the size of the gaze stripes could be too large to deal with, hence, a more scalable variant is needed (Kurzhals et al. 2016a). 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, 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 strengths at different locations of a stimulus and provide an interactive linking to associate such strengths with the actual stimulus content and extra visual representations like visual attention maps (Bojko 2009) and gaze stripes (Kurzhals et al. 2016b). This new approach is based on the well-known word cloud visualization technique (Burch et al. 2013c), but in our case, not showing word frequencies (Khusro et al. 2018), but image thumbnails exposed to different fixation durations. The original work called EyeClouds (Burch et al. 2019c) is extended in this article by supporting image-based clustering (Wang et al. 2015) based on RGB pixel values to detect common patterns among the visually attended regions in a visual stimulus. Moreover, temporal links are integrated as well as further features.

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 patterns among study participants. To reach our goal, we have to compare eye movement data, visualize and order the comparisons, and finally, allow interactions and linkings to the original stimuli.

3.1 Design criteria

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

- Selection of stimuli and viewers The user is able to select the stimuli and the people having participated in the eye tracking study.
- **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 times of the application and waiting times for the user.
- Clear and relevant output The visualization output is clear and relevant to give insights to the user, for space, time, and study participants as well as stimulus information.
- Extend ability and scalability The application is extendable and scalable for different use cases and data sets.

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 3 describes that the



Fig. 3 A high-level diagram of the EyeCloud application



Fig. 4 Front-end design with labeled components: (A) the control panel, (B) a heat map, (C) an attention cloud, and (D) gaze stripes. All components are linked

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. 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 computation times 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. 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 Fig. 4. 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, summarized in the following:

- 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 article, we assume that the data passed to our web application is consistent and free of errors, e.g., we assume 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 have been transformed and integrated inside the application and stored inside the database using a Python 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



Fig. 5 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

683

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: stimulus, stimulus 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. (2016b) for the gaze stripes and by Blignaut (2010) 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, interactively linked to visual attention maps and gaze stripes. Moreover, we provide classical word cloud layouts as well as more image-based clusterings for the extracted image thumbnails. The tool users can select a subset of the eye movement data, and the visualizations are generated rapidly.

As an add-on to the word cloud-based layout, we use image-based clustering since it provides a more structured view on the stimulus content and its characteristics. This is achieved by taking into account image thumbnails that carry visual information about semantics of the displayed stimulus encoded in similar visual features based on color distributions, in our application example on public transport map features (like metro lines). A cluster of image thumbnails can hereby provide insights about a group of similar features that have been visually attended while the thumbnail sizes indicate if certain feature groups are visually attended more or less compared to others.

5.1 Attention cloud

The attention cloud will display images of different cropping sizes with larger ones representing the parts of the stimulus 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 stimulus 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 (Fruchterman and Reingold 1991). This typically generates a space-filling diagram, similar to approaches for word cloud layouts. Even if there are no explicit (visual) links in the diagram, we can apply the force-directed layout by using similarity relations based on the color distributions in each thumbnail image.

In our case, we model the attention cloud as a graph with nodes represented by thumbnails and without (visual) 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 (Fruchterman and Reingold 1991). The application allows the user to change various visualization options like max cropping size, min cropping size, cluster radius, and number of data points.

In Fig. 5, 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.



Fig. 6 The example shows the stimulus being the public transport map of Warsaw with 5 participants selected. The time range is from 0 to 14,439 ms, the minimum and maximum cropping sizes are 20 and 100, the cluster radius is 0: **a** 10 thumbnail images, clustered in **d**. **b** 20 thumbnail images, clustered in **e**. **c** 50 thumbnail images, clustered in **f**

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.

5.1.1 Image-based clustering

Although the force-directed and space-filling layout of the attention clouds is already quite useful to detect visual attention patterns, we enhance the layout by applying image-based clustering. This approach can group image thumbnails that carry similar characteristics, for example, containing similar objects or at least similar color distributions.

The clustering that we apply works quite fast and still allows interaction techniques that are responsive. We are not targeting optimal layouts, but we are trying to achieve good layouts that reflect some structure but, on the other hand, benefit from their low runtime complexities. Typically, there is a trade-off between finding an optimal clustering solution and providing algorithms for that with low runtime complexities.

For the image-based clustering, we first compute the average pixel RGB values for each thumbnail image which is then represented as a three-dimensional vector. K-means clustering is used to assign a cluster number to each thumbnail, based on the number of clusters specified by the user (see Fig. 6 for 10, 20, and 50 thumbnail images).

The layout is realized as follows:

- (1) For each cluster, sort the thumbnails by the radius.
- (2) Layout the thumbnails in a circular manner such that

- (a) the largest thumbnail of each cluster is located at the center of the cluster and surrounded by thumbnails with decreasing sizes, and
- (b) the clusters are located clockwise, with (360/numOfClusters) degrees away from each other.
- (3) Turn on the force simulation to pull thumbnails within a cluster towards the cluster center.

5.1.2 Temporal thumbnail links

The thumbnail images in the attention cloud can be temporally linked. Warm colors reflect latest time points and cold colors indicate time points a while ago in the past. The timestamp of aggregated fixation points is computed by first allocating the fixation timestamps into bins of 1 s and taking the bin of the highest frequency. Also the link widths can be interactively adapted (see Fig. 7 for an example).

It may be noted that the temporal links can lead to visual clutter if a clustered layout is used. The better option is to manually modify certain thumbnail positions, maybe as a linear chain, to identify the temporal sequence of stimulus region visits.

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 stimulus, 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 stimulus and their corresponding thumbnail 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 this article and during the loading phase 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 Fig. 8). The visual span radius integration has been implemented as described by Blignaut (2010) and we also allow the user to view the actual content of the map.



Fig. 7 Each EyeCloud can be temporally linked: Warm colors indicate latest time points while cold colors indicate time points long ago in the past. The color scale uses red as warm color and blue to green as cold colors. The color scale can be interactively adapted



Fig. 8 Different heat map settings for the visual span: a Heat map with visual span at 20px. b Heat map with visual span at 100px

We have added the focus mode as a novel feature in a heat map. This effect shows only the visually attended regions in a stimulus, i.e., the hot spots. Unimportant parts of the stimulus are hence masked. They appear opaquely white and the focused parts in the stimulus are left untouched (see Fig. 9).

5.3 Gaze stripes

The gaze stripes, consisting of a series of thumbnails, will display the scan paths of people who visually inspected a given stimulus. This gives insights into what people pay more attention to within a selected period of time while at the same time providing a contextual information in form of image thumbnails. 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 Fig. 10). However, the temporal links can also be interactively added in the attention clouds, but might cause visual clutter if too many are shown at the same time or the diagram is dense and crowded (Rosenholtz et al. 2005).

Our implementation is based on the one described by Kurzhals et al. (2016b) 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 s of fixation points from a user and the last 3 s 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



Fig. 9 The focus mode only shows the visually attended regions in a stimulus



Fig. 10 Different gaze stripes: a Gaze stripes with scale set at 10px. b Gaze stripes with scale set at 40px

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 Attention bar chart and statistical information

We have also added two new components to get more information about fixation frequencies and textual details. Those views are called attention bar chart and statistics of the selected point. Figures 11 and 12 show examples for these features.

5.5 Interaction techniques

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

- Attention cloud Selecting an image in the attention cloud will show the corresponding point on the heat map, i.e., the views are linked.
- Heat map Clicking a point on the heat map will highlight the corresponding image in the attention cloud and provide extra details.
- Gaze stripes Hovering over a thumbnail on the gaze stripe will highlight the corresponding point in the heat map and in the attention cloud.
- **Data selection** Changing settings such as stimulus, stimulus viewers, or time range will force to recreate the attention cloud, heat map, and gaze stripes.



Fig. 11 A fixation frequency overview for certain thumbnail images

Statistics of the Selected Point	
Point Position	(1279, 314)
Fixation Duration	50841
Fixation Timestamp	8493
Number of aggregated points	135
Raw Timestamps	

Fig. 12 Additional details in form of textual output



Fig. 13 The public transport map of Antwerp is used as a stimulus, all users are selected. 500 thumbnails are generated based on the fixation points with minimum and maximum crop sizes of 20 and 70, respectively. The cluster radius is 0. **a** 3 clusters. **b** 10 clusters. **c** 20 clusters. **d** 100 clusters

- Animation in visual attention map A heat map can be built in an animated fashion in which the animation speed can be interactively modified.
- Adapting clustering parameters The clustering can be guided by modifying several parameters involved, for example the display area or the number of clusters.
- Modifying color codings A default color is selected but users can interactively adapt the color scale to their demands.
- Selecting and changing temporal links Clicking on temporal links gives extra information about the time information and where in the stimulus this happened.
- **Requesting details on demand** Each visual object can be hovered over and details about it can be requested.

When users click on an 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.

6 Application examples

In this section, we will explore a real-world eye movement data set from a formerly conducted eye tracking study investigating the readability of public transport maps. We will visualize it with the attention clouds and describe insights. We have chosen different representative public transport maps with several complexity levels (high, medium, low). We start with the map of Antwerp that provides some insights based on the image-based clustering approach.

6.1 Clustered attention clouds for Antwerp

Figure 13 shows four example attention clouds for the public transport map of Antwerp in Belgium. For the 3 clusters scenario in (a), we see that there is a clear and more or less equally distributed picture of the image thumbnails. The temporal links are not useful in a static picture for 500 thumbnails, but interaction might help here.

For the 10 clusters in (b), we can identify even more visual attention groups based on the image thumbnails. Such groups of similar image characteristics can be used to filter in the stimulus. Moreover, if the temporal information is taken into account, we can see the visual attention changes between several clusters, i.e., if the study participants rather stayed in a certain group or frequently moved between image thumbnail groups.

The 20 clusters (c) and the 100 clusters (d) are already too many to find useful insights. We recommend to interactively adapt the cluster parameter to find the best separation of the image thumbnails into groups.

6.2 New York (high complexity)

New York City (NYC) is the most populous and modern city in the US 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 USA. 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 a 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 Fig. 14.

Firstly, we can get an overall pattern of eye movements from the heat map shown in Fig. 4. 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. We can take a closer look at the attention cloud in Fig. 4.

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



Fig. 14 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



Fig. 15 Gaze stripes of 5 different users for the New York transport map

station and a 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 Fig. 15, 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.

6.3 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 Fig. 16, 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



Fig. 16 Attention cloud (a) and heat map (b) of Warsaw, the biggest thumbnail in the EyeCloud is an intersection point



Fig. 17 Attention cloud (a) and heat map (b) of Venice, two of the three biggest points in the attention cloud are intersection points

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.

6.4 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. We can 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 Fig. 17): 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 Discussion and limitations

As shown in the application examples, we could partially prove that our tool is able to provide solutions to the research questions RQ1 to RQ3. However, for proving the usefulness of RQ4, we need to conduct a user experiment with non-experts in visualization. By letting students from a visualization course interactively experiment with the visualizations, we got first hints that it can be understood and used. For a more thorough study, we have to record task completion times, error rates, or even eye movements as well as qualitative feedback. This means its practical usefulness needs further validation, for example, when compared with other types of views, such as zoomed in map views or hierarchically clustered map views.

In the EyeCloud approach, the data analysts are able to select as many map viewers as possible. This feature allows the analysts to select, visualize, and analyze any subset of the eye movement data to fulfill their needs. The application should be able to respond in a few seconds. However, when the application is confronted with increasing input size, this is no longer the case. Some of the described functionalities are subject to scalability issues, for example, the force-directed placement in the attention clouds.

One of the challenges is the capacity of the server where the application is running on. As the size of eye movement data increases, the queries from MongoDB will take much longer time, and the back-end computations to aggregate and process selected data will also cost more resources. Hence, the response time from the server to the client will increase drastically.

Another main challenge is the gaze stripes. As mentioned before, the gaze stripes have limitations for visualizing a large data set. In the EyeCloud, the way we choose to present the gaze stripes is to fix the smallest number of pixels of a thumbnail to maintain the information. The issue is that if the data set becomes large, it is tedious for the analysts to trace the scanpath of every map viewer and find the common patterns of eye movement behaviors. However, such an issue is not severe as it may seem because the main visual components that give quick and overall information of the data are the attention cloud and the heat map where the analysts are able to gain most insights from them. Gaze stripes only provide additional details of the eye movement behavior of each individual map viewer, the analysts are expected to spend more time on this visual component to gain further information.

Furthermore, as the tool is a web application, not all of the issues can be solved by increasing the capacity of the server it is running on. A lot of tasks are performed by the users' browser and rely on local computational resources. For example, constructing an attention cloud, a heat map, and gaze stripes with thousands of fixation points will considerably decrease the applications' responsiveness, on an average notebook. Thus, it is reasonable to decrease the interactivity of the application by putting a limit on the resolution of the visual components and selection of maps and participants. In fact, if inputs would be well balanced with the available resources, these scalability issues are not severe as they may seem. A more experienced user might navigate through large input by avoiding too demanding data selections and parameter settings. Letting the availability of certain functionality depend on the available resources would greatly increase the scope of the tool.

For the image-based clustering, it is important to interactively modify the number of clusters to see which one provides the best results. Moreover, the temporal links can be useless if too many thumbnails are shown. In such a scenario, it is beneficial to apply interaction techniques to filter the number of displayed elements.

8 Conclusion and future work

We have created a web application, the EyeCloud, in which the user can select a stimulus with fixation data of multiple viewers to gain an overview of common visual attention patterns and distributions of visual attention strength. This can be linked to locations on the stimulus for contextual information, to compare scanpaths between participants, while the results are depicted in an attention cloud, a heat map, and gaze stripes. In this article, we enhanced the standard word cloud-based and space-filling layout by image-based clustering and temporal links. Moreover, we added a way to only show the visual attended regions in a stimulus and provide views about visual attention frequencies shown in bar charts and details-on-demand with statistical information. Also many more interaction techniques are integrated. The visual components can be reconstructed by selecting a different subset of stimuli and viewers and are interactively responsive. The major goal was to create a tool for supporting eye movement data analysts to visualize and explore eye movement data. For future work, there are several ways that could improve this application. Firstly, it would be beneficial to give the analysts more options to include more information from eye tracking experiments like pupil dilations, galvanic skin response, or even EEG data. Possible options are also additional descriptive statistics of the stimuli and viewers linked to the interactive visualizations. It should also be possible to better compare two or more stimuli and to select subsets of viewers and paths in all of them. Furthermore, this visualization tool might be extended to many other purposes with various types of data, for example, general trajectory data.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

- Blascheck T, Burch M, Raschke M, Weiskopf D (2015) Challenges and perspectives in big eye-movement data visual analytics. In: Proceedings of the 1st international symposium on big data visual analytics, pp 17–24
- Blascheck T, Kurzhals K, Raschke M, Burch M, Weiskopf D, Ertl T (2017) Visualization of eye tracking data: a taxonomy and survey. Comput Graph Forum 36(8):260–284
- Blignaut PJ (2010) Visual span and other parameters for the generation of heatmaps. In: Proceedings of the 2010 symposium on eye-tracking research and applications, ETRA, pp 125–128
- Bojko A (2009) Informative or misleading? Heatmaps deconstructed. In: Human-Computer Interaction–INTERACT. Springer, Berlin, pp 30–39
- Burch M (2016) Time-preserving visual attention maps. In: Proceedings of conference on intelligent decision technologies, pp 273–283
- Burch M (2017a) Mining and visualizing eye movement data. In: Proceedings of SIGGRAPH ASIA symposium on visualization, pp 3:1-3:8
- Burch M (2017b) Visual analysis of eye movement data with fixation distance plots. In: Proceedings of conference on intelligent decision technologies, pp 227–236
- Burch M (2018) Identifying similar eye movement patterns with t-SNE. In: Proceedings of vision, modeling and visualization, VMV, pp 111–118
- Burch M, Andrienko GL, Andrienko NV, Höferlin M, Raschke M, Weiskopf D (2013a) Visual task solution strategies in tree diagrams. In: Proceedings of IEEE Pacific visualization symposium, pp 169–176
- Burch M, Kull A, Weiskopf D (2013b) AOI rivers for visualizing dynamic eye gaze frequencies. Comput Graph Forum 32(3):281–290
- Burch M, Lohmann S, Pompe D, Weiskopf D (2013c) Prefix tag clouds. In: Proceedings of international conference on information visualisation, IV, pp 45–50
- Burch M, Kurzhals K, Weiskopf D (2014) Visual task solution strategies in public transport maps. In: Proceedings of ET4S@GISCIENCE, pp 32-36
- Burch M, Kumar A, Mueller K (2018a) The hierarchical flow of eye movements. In: Chuang LL, Burch M, Kurzhals K (eds) Proceedings of the 3rd workshop on eye tracking and visualization, ETVIS@ETRA. ACM, New York, pp 3:1–3:5

- Burch M, Kurzhals K, Kleinhans N, Weiskopf D (2018b) EyeMSA: exploring eye movement data with pairwise and multiple sequence alignment. In: Proceedings of the 2018 ACM symposium on eye tracking research and applications, ETRA, pp 52:1–52:5
- Burch M, Kumar A, Mueller K, Kervezee T, Nuijten W, Oostenbach R, Peeters L, Smit G (2019a) Finding the outliers in scanpath data. In: Proceedings of the 11th ACM symposium on eye tracking research and applications, ETRA, pp 83:1–83:5
- Burch M, Kumar A, Timmermans N (2019b) An interactive web-based visual analytics tool for detecting strategic eye movement patterns. In: Proceedings of the 11th ACM symposium on eye tracking research and applications, ETRA, pp 93:1–93:5
- Burch M, Veneri A, Sun B (2019c) EyeClouds: a visualization and analysis tool for exploring eye movement data. In: Proceedings of the 12th international symposium on visual information communication and interaction, VINCI. ACM, pp 8:1–8:8

Duchowski AT (2003) Eye tracking methodology-theory and practice. Springer, Berin. ISBN 978-1-85233-666-0

- Fruchterman TMJ, Reingold EM (1991) Graph drawing by force-directed placement. Softw Pract Exp 21(11):1129–1164 Goldberg JH, Helfman JI (2010) Visual scanpath representation. In: Proceedings of the symposium on eye-tracking research and applications (ETRA), pp 203–210
- Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, van de Weijer J (2011) Eye Tracking: a comprehensive guide to methods and measures. Oxford University Press, Oxford
- Keim DA (2012) Solving problems with visual analytics: challenges and applications. In: Proceedings of machine learning and knowledge discovery in databases—European conference, pp 5–6
- Khusro S, Jabeen F, Khan A (2018) Tag clouds: past, present and future. In: Proceedings of the national academy of sciences, India section A: physical sciences, pp 1–13
- Kumar A, Timmermans N, Burch M, Mueller K (2019) Clustered eye movement similarity matrices. In: Proceedings of the 11th ACM symposium on eye tracking research and applications, ETRA, pp 82:1–82:9
- Kurzhals K, Hlawatsch M, Burch M, Weiskopf D (2016a) Fixation-image charts. In: Proceedings of the ninth biennial ACM symposium on eye tracking research and applications, ETRA, pp 11–18
- Kurzhals K, Hlawatsch M, Heimerl F, Burch M, Ertl T (2016b) Weiskopf D (2016b) Gaze stripes: image-based visualization of eye tracking data. IEEE Trans Vis Comput Graph 22(1):1005–1014
- Munzner T (2014) Visualization analysis and design. A K Peters, Natick. ISBN 978-1-466-50891-0
- Netzel R, Ohlhausen B, Kurzhals K, Woods R, Burch M, Weiskopf D (2017) User performance and reading strategies for metro maps: an eye tracking study. Spat Cogn Comput 17(1–2):39–64
- Rosenholtz R, Li Y, Mansfield J, Jin Z (2005) Feature congestion: a measure of display clutter. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 761–770
- Tufte ER (1992) The visual display of quantitative information. Graphics Press, Cheshire. ISBN 978-0-9613921-0-9
- Wang J, Wang J, Song J, Xu X-S, Shen HT, Li S (2015) Optimized Cartesian K-means. IEEE Trans Knowl Data Eng 27(1):180–192
- Ware C (2008) Visual thinking: for design. Morgan Kaufmann Series in Interactive Technologies, Paperback
- Yi JS, Kang Y, Stasko JT, Jacko JA (2007) Toward a deeper understanding of the role of interaction in information visualization. IEEE Trans Vis Comput Graph 13(6):1224–1231

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.